

ABSTRACT

Title of Thesis: ORDER ASSIGNMENT AND RESOURCE
RESERVATION: AN OPTIMIZATION MODEL AND
POLICY ANALYSIS

Degree candidate: Julie Tricia McNeil

Degree and year: Master of Science, 2005

Thesis directed by: Professor Michael Ball
Decision and Information Technologies Department
R.H. Smith School of Business
Institute for Systems Research

To maintain a competitive edge, companies today must be able to efficiently allocate resources to optimally commit and fulfill requested orders. As such, order processing and resource allocation models have become increasingly sophisticated to handle the complexity of these decisions. In our research, we introduce a model that integrates production scheduling, material allocation, delivery scheduling, as well as functions involving commitment of forecast demand for configure-to-order (CTO) and assemble-to-order (ATO) business environments. The model is formulated as a Mixed Integer Program (MIP) and seeks to maximize revenue by trading off commitment of higher profit forecast orders with the production and delivery schedule of lower profit accepted orders. Our model is

particularly useful for testing different policies relating to order commitment, delivery mode selection and resource allocation.

**ORDER ASSIGNMENT AND RESOURCE RESERVATION:
AN OPTIMIZATION MODEL AND POLICY ANALYSIS**

By

Julie Tricia McNeil

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2005

Advisory Committee:

Professor Michael Ball, Chair
Professor Zhi-Long Chen
Professor Mark Austin

© Copyright by

Julie Tricia McNeil

2005

ACKNOWLEDGEMENTS

I would like to thank Dr. Michael Ball and Dr. Zhenying Zhao for their guidance and support in the development of this research. I truly appreciate their advice and constructive feedback over the course of my work at the University of Maryland.

I also want to thank my family and friends for their continued support over the past two years.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS.....	iii
LIST OF TABLES	v
LIST OF FIGURES	vii
1. Introduction.....	1
1.1. Research Objectives.....	3
1.2. Organization of Thesis.....	4
1.3. Literature Review.....	5
1.3.1. Planning Models	6
1.3.2. Order Promising.....	8
1.3.3. Revenue Management.....	9
1.3.4. Optimization-based DSS for Production Planning	11
2. Optimization Model	13
2.1. Overview of Business Application of Model.....	13
2.1.1. Products.....	15
2.1.2. Factories	17
2.1.3. Merging Centers.....	17
2.1.4. Transportation Modes for Order Delivery	18
2.1.5. Orders and Demand	18
2.2. Performance Measures.....	19
2.3. Business Policies.....	20
2.3.1. Service Levels.....	20
2.3.2. Commitment	24
2.4. Assumptions.....	25
2.5. Model Formulation	27
3. Model Implementation.....	35
3.1. System Architecture Design	35

3.2.	Software Implementation.....	37
3.2.1.	Use of Xpress.....	37
3.2.2.	Use of Excel.....	38
3.2.3.	Interaction of Xpress and Excel.....	41
3.3.	Data Setup.....	44
3.3.1.	Model Size	44
3.3.2.	Data Generation for Experiments	45
3.3.3.	Generation of Data.....	51
3.4.	Model Specifications	55
4.	Study of Experiments.....	58
4.1.	Verification of Model	58
4.2.	Sensitivity Analysis	60
4.2.1.	Base Setup of Model.....	61
4.2.2.	Sensitivity Analysis of Profit Margins.....	63
4.2.3.	Sensitivity Analysis of Production Capacity	66
4.2.4.	Sensitivity Analysis of Part Shortage (Unique Part).....	70
4.2.5.	Sensitivity Analysis of Part Shortage (Common Part)	72
4.3.	Experiment 1: Commitment Policy	74
4.4.	Experiment 2: Service Level Policy Analysis	78
4.5.	Experiment 3: Re-pointing of Merging Center.....	83
4.6.	Experiment 4: Re-pointing of Production Capacity.....	97
5.	Conclusion	102
5.1.	Summary of Results.....	102
5.2.	Future Work.....	103
	Appendix A: Xpress Mosel Code	105
	Appendix B: Selected Excel VB Code	113
	References.....	116

LIST OF TABLES

Table 3.1: Model Indexes	39
Table 3.2: Parameters and Associated Indexes.....	40
Table 3.3: Dataset Size of Parameters	45
Table 3.4: Formulation of Shipping Costs per Transportation Mode	46
Table 3.5: Revised Formulation of Shipping Costs per Transportation Mode	47
Table 3.6: Formulation of Profit Margins.....	49
Table 3.7: Formulation of Parameter Values.....	54
Table 3.8: Model Size – Xpress Solver	55
Table 3.9: Comparison of Computational Complexity of Two Runs	57
Table 4.1: Range of Values for Parameters in Base Model.....	62
Table 4.2: Results of Base Trial.....	62
Table 4.3: Parameter Settings for Sensitivity Analysis of Profit Margins.....	64
Table 4.4: Results of Sensitivity Analysis of Profit Margins	65
Table 4.5: Parameter Settings for Sensitivity Analysis of Production Capacity	67
Table 4.6: Results of Sensitivity Analysis of Production Capacity	68
Table 4.7: Profit Margin Results of Sensitivity Analysis of Production Capacity ..	69
Table 4.8: Results of Sensitivity Analysis of Part Shortage (Unique Part)	71
Table 4.9: Results of Sensitivity Analysis of Part Shortage (Common Part)	73
Table 4.10: Profit Margin Results of Sensitivity Analysis of Part Shortage (Common Part).....	74
Table 4.11: Results of Experiment 1: Commitment Policy	76
Table 4.12: Analysis of Effect on Orders for Commitment Policy Experiment.....	77
Table 4.13: Results of Experiment 2: Service Level Policy	80
Table 4.14: Results of Experiment 2: Service Level Policy (Equal Objective Weights).....	82
Table 4.15: Results of Experiment 3: Merging Center Re-Pointing.....	90
Table 4.16: Comparison of Results by SKU and Order/Demand.....	92

Table 4.17: Summarized Results of Merging Center Re-Pointing	92
Table 4.18: Effect of Merging Center Re-pointing on Orders/Demand	93
Table 4.19: Results of Experiment 4: Production Center Re-pointing	98
Table 4.20: Comparison of Results of Production Re-pointing for Single Trial (Same Data)	99
Table 4.21: Kitting SKUs Sorted by Profit Margin	99

LIST OF FIGURES

Figure 2.1: Overview of the Major Decisions in Supply Chain included in Model	15
Figure 2.2: Definition of Kitting and Merging SKUs.....	16
Figure 2.3: Service Level Overview	23
Figure 3.1: High-level System Architecture	36
Figure 3.2: Details of Xpress-MP Suite.....	38
Figure 3.3: Required Data for Orders and Demand.....	41
Figure 3.4: Interaction of Excel and Xpress	43
Figure 3.5: Graph of Shipping Costs per Transportation Mode	47
Figure 3.6: Chart of Profit Margins per SKU and Service Level	50
Figure 3.7: Branch and Bound Search to Identify MIP Solution.....	56
Figure 4.1: Lead Times from Merging Centers to Customer Regions	90
Figure 4.2: Chart of Delivery Quantities from each Merging Center.....	95
Figure 4.3: Chart of Due Date Penalties per Merging Center.....	95
Figure 4.4: Chart of Delivery Quantities per Service Level	96
Figure 4.5: Chart of Revenue Differences across the Product/Service Level Configurations.....	100

1. Introduction

In today's customer-driven marketplace, companies are held to higher standards and expectations in the management of their supply chain. Industry leaders are using supply chain expertise as a main selling point in their business models – Amazon.com woos customers with a seemingly endless supply of merchandise options, almost all of which are available for immediate shipping (see Bachelder 2004, Andel 2000) and Dell offers customizable product configurations shipped direct with next-to-nothing lead times (see Buderer 2001). Supply chain management capabilities relate to how well a company can procure supplies, manage inventory, schedule production, package products, deliver orders and process customer requests, among other functions. Successful ATO/CTO facilities use highly developed models for scheduling and planning their supply chain. By using an ATP mechanism in the planning model, the production scheduling can be linked to the order promising function. Essentially, the ATP capability first determines whether capacity exists to fulfill the incoming order request, and then determines the corresponding due date and quantity that can be promised for accepted orders. By associating the order commitment process with the production resource allocation, better decisions can be made for ATO/CTO companies.

An area of increased focus is the definition of the commitment policy in the ATP model for incoming orders, which is especially relevant for companies with greater demand than capacity. The standard policy in place at many facilities is

that of a *First-come, First-served* commitment policy. Customer orders are accepted in the sequence in which they arrive until all capacity has been allocated, at which point any additional incoming orders are denied. This policy promises fairness to the customer, but in practice does not favor loyal customers and more importantly, does not capitalize on the greater potential revenue of some orders over others. Two contemporary policies for commitment include methodology to discriminate between new orders based on the principles of revenue management. The first policy uses the concept of customer channels in the commitment decision process. This is standard with many service industries today (e.g. for an airline, a fraction of the capacity (seats) is reserved for potential late-booking, higher revenue passengers (see Smith et al, 1992)). The second policy is similar, but bases the commitment decisions on the relative profit margin of the incoming orders. Thus, resources are reserved for the orders with the greatest contribution to overall revenue first.

These commitment policies lead directly into the issue of trying to balance reservation of capacity for accepted orders and for forecast demand. Once orders have been committed, the manufacturer cannot renege on the order simply because a higher profit order came in. Within the advanced ATP model, consideration can be given to future demand of orders so that capacity can be reserved accordingly. An additional way to add flexibility to the reservation of resources is to allow some orders to be delivered late, for a penalty. This creates additional capacity for the commitment of higher profit orders. Due date violations are not desirable, but their

use can help mitigate the loss of profits due to order rejection in certain situations. A new strategy is to consider trading off delivery mode and schedule with the production schedule. In this approach, the company can delay production of some lower profit orders so that capacity can be allocated to higher profit demand. Instead of delivering the orders late, the company can upgrade the delivery mode so that the order arrives to the customer by the requested date. Any extra transportation fees would be balanced by the extra profits.

We can see that the need exists for a comprehensive model that integrates the order and demand promising functions with resource reservation analysis. The development of such a model would enable full analysis of any new policies for commitment and delivery scheduling.

1.1. Research Objectives

The first objective of this thesis is to develop an integrated model for order promising and resource allocation that trades off production efficiency and delivery scheduling. To accomplish this, we must first define the general business scenario. Next, we determine the policies and goals of our system and formulate it as a mixed integer program. Our motivation is twofold – we want to create an enhanced model that considers profit contributions in the commitment decisions of demand and resource allocation, and we want the model to balance the production schedule with the delivery schedule and mode choice for orders.

After developing the model, our next objective is to prove its capabilities in order assignment and production planning. We want to show that our model can be

used as a powerful decision support system for supply chain managers. This is done through examples of how policies can be tested and analysis of the results. As part of this objective, we want to show the power of using a spreadsheet-based front end model. The goal is to prove that it can efficiently handle problems of standard size and is sufficiently simple to understand and use, by even non-modelers.

1.2. Organization of Thesis

In Chapter One, the research objectives are introduced, including background on the research problem, and a summary of our research contributions. This is followed by a summary of research in this area, through a comprehensive literature review focusing on resource planning, ATP mechanisms and order promising, revenue management, and the use of decision support systems in production planning.

Chapter Two provides a general overview of the optimization model that we developed. The model is explained from a business-perspective, including information regarding the manufacturing setup, product definition, and other aspects of the model. The major decisions, assumptions, policies, and performance measures of the model are addressed. Finally, the chapter presents the mathematical formulation of the model. All parameters, indexes and decision variables are defined for the mixed integer program. Additionally, the objective function and constraints are given, with a detailed text description of each.

In Chapter Three, we cover the implementation of the model. Discussion of the software selection, capability and integration is provided. This is followed by a brief discussion of the model specifications, including computational analysis. Finally, we address the area of input data for the model; essentially, how data was developed to run the various trials.

The next section, Chapter Four, delves into the analysis of the model. To begin, the process of model verification is described. This is followed by the sensitivity analysis, in which the model capabilities are explored by varying parameters and business scenarios. The results of these tests are analyzed fully. Next, we discuss the various experiments that were conducted. We provide a discussion of how the experiment affects the model, in terms of any changes in business policies that alter the model formulation. Each experiment's purpose is discussed, followed by a review of the results.

Finally, in Chapter Five, we summarize the results and findings of our research. We also present areas for future research.

1.3. Literature Review

Four areas of research that are closely related to our research are: resource planning (forecast-based and resource allocation), ATP mechanisms and order promising, revenue management (including resource booking based on due date), and use of optimization-based decision support systems in production planning.

1.3.1. Planning Models

Understandably, the bulk of early research in this area focuses on effective methods for production planning, scheduling, and inventory control. Johnson and Montgomery (1974) were among the first to develop generic mixed integer models for these applications. With the increased growth of configure-to-order (CTO) and make-to-order (MTO) production settings, more sophisticated analysis models or tools are needed to handle the complexity of resource allocation, production scheduling, and order assignment. McClelland (1988) focuses on using the Master Production Schedule (MPS) in order management models. In practice, the MPS is developed from the aggregate production plan, inventory stock, and accepted orders. The selection of an appropriate MPS system can lead to more efficient capacity and material allocation, resulting in an increase in service performance measures (including the ability to keep a higher fraction of order due date promises).

The Available-to-Promise (ATP) model evolved from these earlier models. Fundamentally, the ATP mechanism links various production and delivery resources and order processing; it determines order promising (both due date and quantity) and order fulfillment (production scheduling) based on resource availability. Vollman, Berri, and Whybank (1997) provide a comprehensive overview of conventional ATP models in their book.

Much of the recent research of ATP focuses on enhancements to the standard ATP system in regards to order promising capabilities. Greene (2001) stresses the importance of ATP systems in which current demand, production

problems and supply constraints are kept visible. He further discusses the need for new technology to determine optimal promise dates for fulfillment efficiencies and profitable-to-promise (PTP) outcomes. Taylor and Plenert (1999) generate a heuristic called Finite Capacity Planning, in which the production schedule is analyzed to identify unused capacity. This in turn enables more realistic setting of order promise dates for customer orders. Hariharan and Zipkin (1995) research new methods to improve how inventory is modeled. Specifically, they analyze how advance information of customer orders affects inventory policies. Their stochastic model focuses on procurement efficiency over resource utilization.

Dhaenens-Flipo and Finke (1999) created a model that includes analysis of multiple factories and multiple products over a rolling timeframe. They enhance the traditional model by studying both the production and distribution functions in the creation of production schedules. The resulting schedule is based on minimization of holding costs, production costs, changeover costs, and transportation costs.

Chen, Zhao and Ball (2001) introduce an optimization-based ATP model that takes into account the current status of the production system and can dynamically allocate and reallocate material and capacity. Thus, the profitability of orders can be traded off. Their MIP model enables such features as order splitting, model decomposition and resource expedition/de-expedition. It determines both the quantity and due date quotes for orders. Our model uses this model as its foundation.

1.3.2. Order Promising

Order promising is complex because of the general lack of inventory in an ATP MTO/CTO environment. In response, several papers have been written which focus solely on due date scheduling models. The models can be divided into two groups – those with exogenous due dates and those with endogenous due dates. Exogenous due dates refer to those settings in which the customer determines the due date. Endogenous due dates refer to those settings in which the manufacturer supplies the due date to the customer (see Cheng and Gupta (1989) for a detailed survey). In endogenous settings, the manufacturer must trade off offering short lead times of delivery quotes (potentially increasing customer demand) and meeting those due dates reliably (failure to do so results in customer dissatisfaction).

Hegedus and Hopp (2001) focus on due date quoting in a make-to-order environment in which the customer requests a specific due date. The manufacturer can then accept the due date or provide an alternative later date. Chatterjee, Slotnick and Sobel (2001) develop a profit maximization model with endogenous due date assignments. In their model, the customer can accept or reject the potential due date (“balking”). Additionally, they allow orders to be delivered late (for a cost). See also Hopp and Roof Sturgis (2000) and Hopp and Sturgis (2001) for additional endogenous due date setting models.

Moses, et al (2004) introduce a model which has real-time promising of order due dates, applicable to make-to-order environments. The model considers

availability of resources, order specifics, and existing commitments to orders that arrived earlier. The order arrival rate is stochastic, and orders can be delivered late. Moodie and Bobrowski (1999) merged the two classifications of due dates. In their model, the customer and manufacturer negotiate the due date and the price of an order.

Our model is a blend of the exogenous and endogenous due date policies. While the customer chooses the due date for an order, he must select from a set of available dates provided by the manufacturer, each with an associated cost. The model then determines if it will reserve resources for this demand, based on the production schedule and profit margin contribution of the order. Additionally, the model can deliver orders late for a penalty.

1.3.3. Revenue Management

Further enhancements to the order assignment model involve the consideration of profits. No longer are manufacturers determining the production scheduling and due date assignments based solely on resource capacity or customer service levels. In the situation where demand is much greater than supply, the manufacturer must reject some of the incoming orders. By using revenue management policies, the overall profits can be maximized by selectively choosing which orders to accept (compared to a first-come, first-served basis).

Kirche, Kadipasaoglu, and Khumawala (2005) describe this new model of ‘profitable-to-promise’ order management. In an MTO environment, both capacity and profitability are considered. Specifically, they study which costs are relevant

and how they should be considered in the decision process. Much of the research scope is on using Activity-based Costing (ABC) with Theory of Constraints (TOC) methodology. They develop an MIP model to determine order commitment. Each order is evaluated in terms of profitability to determine if it should be accepted or rejected. Orders that are accepted must be delivered on-time; a penalty applies when orders are rejected.

Balakrishnan, Sridharan, and Patterson (1996) apply a decision theory-based approach to revenue management of order assignment. They develop a capacity allocation policy for discrimination of two demand classes of products (varied profit margins). Akkan (1997) develops heuristics to minimize the cost of rejecting orders. He focuses on reserving future capacity for arriving orders using finite-capacity scheduling-based production planning, where the goal is to minimize contribution lost from rejecting orders. He aims to satisfy customer demand by allocating resources so that revenue and profitability are optimized. Barut and Sridharan (2005) study demand-capacity management policy in an order-driven production system. They develop a heuristic for dynamic capacity allocation procedure that discriminates between incoming orders based on relative profit margins. They use decision tree analysis to determine whether an incoming order should be accepted or rejected. With their policy, the model tends to reject all large quantity orders of the lower profit margins at the beginning of the model, in anticipation of higher profit demand (which may or may not come in). Clearly,

with this greedy policy, the model is likely to reject all low profit orders in a scenario of tight capacity.

Our model uses revenue management in a similar fashion as these models – incoming orders are evaluated based on profit margins. When the capacity is limited, resources are allocated to high profit demand first. The model seeks to maximize overall revenue, and as such, favors higher profit margin orders over lower profit margin orders.

1.3.4. Optimization-based DSS for Production Planning

We next shift our attention to the review of literature discussing decision support systems (DSS) for production planning. Plenert (1992) provides a general survey of the uses of DSS in manufacturing, including applications in forecasting, aggregate production scheduling, finished goods distribution, MPS, MRP, and capacity planning.

We are interested specifically with the use of spreadsheet-based DSS in which an optimization model is used for the analysis. Traditionally, supply chain DSS's were designed by information systems groups, and used by modeling experts. We can see a noticeable shift in this trend – managers today are much closer to the decision process and are more likely to need a DSS in their daily job functions. As such, models have been developed using Excel or other spreadsheet software. These models can be run efficiently by the users, and serve as valuable DSS. See Coles and Rowley (1996) and Troutt (2005).

Smith (2003) adds additional insight into the benefits of spreadsheet models for analyzing logistics and supply chain issues. Namely, they allow analysis from different perspectives and can be modified and enhanced easily. In most cases, an integrated spreadsheet model consists of the baseline model (current / as-is state), the new scenario, and the analysis section to compare the alternatives. Sophisticated software packages are not always needed; spreadsheet models can provide the appropriate level of detailed analysis and realistic/workable solutions. As proof, see papers by Butler and Dyer (1999), Katok and Ott (2000), and LeBlanc et al (2004) for examples of practical DSS involving spreadsheet modeling using optimization programs.

We chose to use a spreadsheet application for the front end of our model and an optimization solver at the back end due to the past successes in similar applications.

2. Optimization Model

In this chapter, we provide an overview of the business application of the model. We then define how the model encompasses the various business settings of the MTO/CTO environment in its analysis. Finally, we give the model formulation.

2.1. Overview of Business Application of Model

In our research, we set out to create a model that would serve two major functions: order promising and resource scheduling (booking), including both assembly and transportation resources for current accepted orders and future higher profit demand. By integrating these two areas, we generate not only an order promising plan, but also the assembly and delivery schedule optimally. Furthermore, our model allocates resources by considering two types of demand – those orders that have been committed, and those orders that are forecast for the future. Thus, it determines when capacity should be reserved for high profit forecast demand, thereby shifting the production schedule of lower profit committed orders. In addition to production capacity, the model also determines the allocation of transportation resources, by determining the schedule and mode for delivery of each order. As such, the model trades off production and delivery schedules of accepted low profit margin orders with higher profit margin demand.

Undoubtedly, the model's power lies in determining the scheduling of production and delivery of both accepted orders and forecast demand.

The model we developed is for companies with an ATO/CTO business setting, and it analyzes the supply chain from order processing through production, assembly and delivery to the customer. Using a Mixed Integer Programming (MIP) model, we determine commitment levels for both committed orders and forecast demand. Furthermore, the detailed product assembly schedule and inventory levels at each factory are determined based on the committed orders and demand. Additionally, the inventory levels and merging schedules are established for the merging centers. Finally, the model determines the delivery schedule and mode for each order for transport to the customer. The following diagram provides a pictorial of the model application and the major decisions.

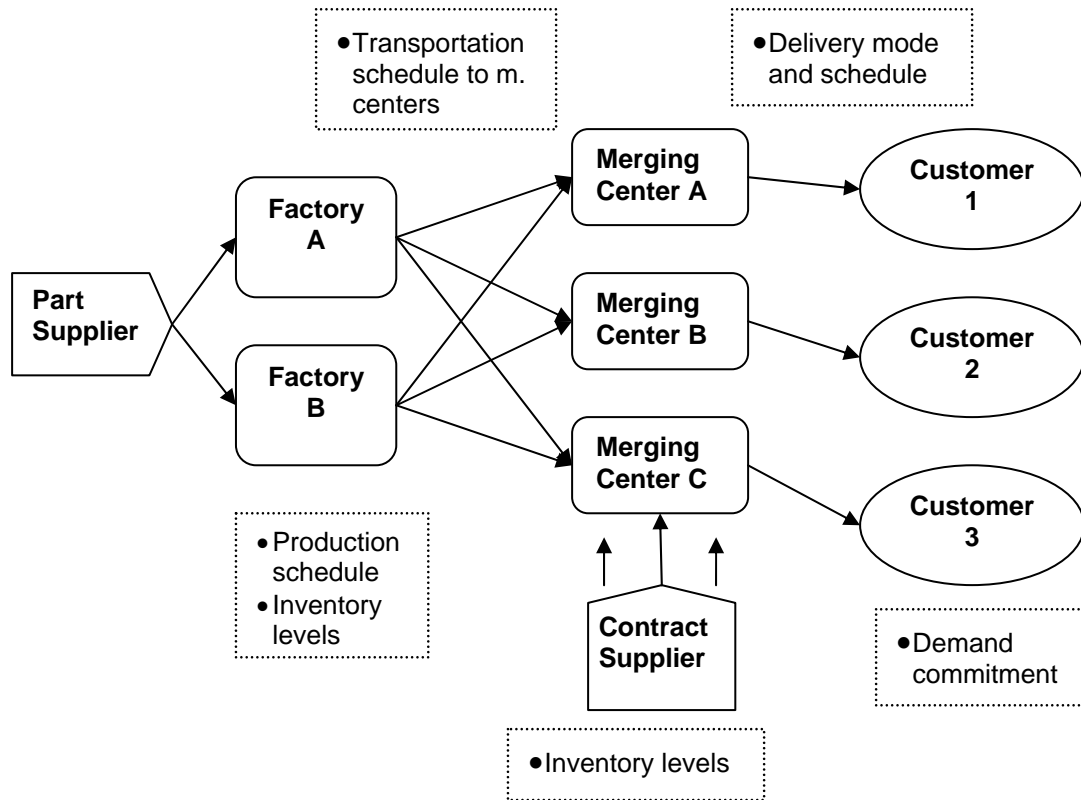


Figure 2.1: Overview of the Major Decisions in Supply Chain included in Model

The following sections define in further detail the various business areas incorporated in the model.

2.1.1. Products

In the supply chains of the ATO/CTO facilities used for our business scenario, the customer orders are comprised of various Stock Keeping Units (SKUs). We classify these SKUs as either Kitting SKUs (kSKUs) or Merging SKUs (mSKUs). The Kitting SKUs are produced in-house by the manufacturer at its factories, while the Merging SKUs are produced by contract suppliers and are shipped directly to the merging centers for final order assembly. This added flexibility is useful in accurately describing the set of products offered by a

company. Each Kitting SKU has a related Bill of Materials (BOM) that details the parts needed for assembly. These BOMs are fixed for each Kitting SKU. The following diagram provides an overview of how the Kitting and Merging SKUs are defined.

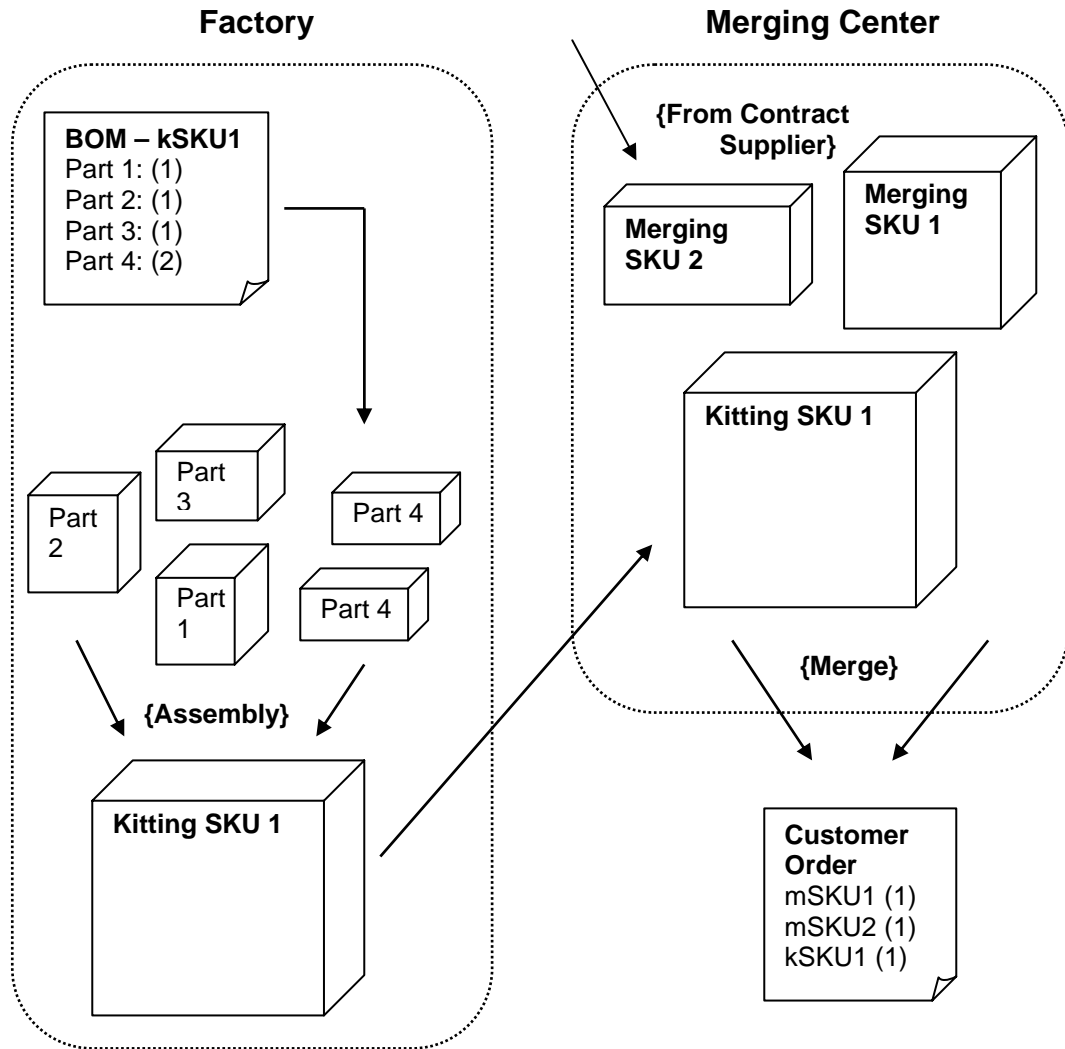


Figure 1.2: Definition of Kitting and Merging SKUs

Finally, each Kitting SKU and Merging SKU has an associated profit margin. The profit margins for each product are further categorized by the service level affiliated with each order, as described later in more detail.

2.1.2. Factories

In the business scenario for the model, multiple factories each with associated daily production capacity are considered. These factories assemble the Kitting SKUs. As mentioned earlier, each Kitting SKU has an affiliated BOM that details the parts needed for assembly. The factories receive daily shipments of the needed parts, that are then stored as inventory until used in production. Ideally, the inventory levels of parts are kept low by accurately forecasting the level of parts needed for each day of production. The stock of parts shipped to each factory on a daily basis is variable, to account for differences in each factory's capabilities.

2.1.3. Merging Centers

As part of the supply chain, our research assumes that the assembled Kitting SKUs are transferred to a merging center, where they then are packed with any associated Merging SKUs to complete each order and are delivered to the customer. In this business scenario, multiple merging centers are used, thereby saving final shipping costs of orders to customers due to the larger network of distribution points. The merging centers serve customers based on geographic proximity; when an order comes in, it is assigned to the closest merging center.

2.1.4. Transportation Modes for Order Delivery

One important decision in the model is the shipment choice for each promised order; essentially how and when an order will be shipped from the merging center to the customer. In our model, an order can be delivered to the customer via one of three transportation modes, depending on the urgency required and the associated costs for each delivery type. The mode refers to the transportation type, such as 1-Day Air or 3-Day Ground. Each mode has a related lead time for delivery. Additionally, it has a flat fixed fee per order as well as a variable fee based on the order volume.

2.1.5. Orders and Demand

As mentioned earlier, the model deals with two classes of orders: those that have already been committed, and those that are forecast for the future. In the terminology used for this model, orders represent actual customer requests with a particular configuration of products, a desired quantity, the requested service level and the location. The forecast demand, meanwhile, is a prediction of orders that will come in the following day. Like orders, each forecast demand has a product configuration, predicted quantity and service level. However, no geographic location is assigned for demand forecasts since this is specified for the whole supply chain.

For the model, we use aggregate values for the orders and forecast demand. Thus, each particular order represents all the orders of that specific configuration and location.

2.2. Performance Measures

As discussed earlier, the model determines the order commitment quantity and corresponding assembly schedule and the delivery schedule. These decisions are made with the overall goal to maximize total profits. Hence, the objective function is broken down into three major parts: revenue, costs and due date violation penalties. Each of these can be weighted to reflect the goals of the company. For instance, some companies might want to avoid delivering orders late whenever possible. By giving the due date violation a higher weight, the model seeks to minimize that term (as it is negative) to maximize the overall revenue.

The first term of the objective function is the revenue associated with the orders and committed demand. The profit for each SKU is defined as the profit margin, or the amount the company would net after production costs, material costs and other associated overhead costs. These profits are calculated upon delivery; as such, orders that are not delivered within the model time frame (in the case of extremely late orders) do not have associated profits. The profit margins vary for each product and each of the service levels available with each product. However, each product has the same profit margin regardless of whether it is associated with an order or a forecast demand order. Higher weights can be assigned to the profits of orders over demand to reflect a preference of known orders and revenue over uncertain demand revenue.

The second term of the objective function is comprised of the delivery costs of orders and committed demand. In the pricing model used, the profit margin of a

SKU is based solely on the service level chosen by the customer. And, with the new policy (to be discussed in the section), transportation modes are not directly associated with service levels. As such, they must be considered separately. We can see how this enables the model to trade-off delayed production with a faster delivery (more expensive delivery costs) while reserving production for higher profit margin demand (resulting in a net gain, assuming the added profits from the demand are greater than the extra delivery costs for the expedited shipping).

The final term in the objective function is the due date violation, which is a penalty assessed for late delivery of orders. Obviously, a penalty must be imposed when orders are delivered after their assigned date. If not, the model would accept all future demand and deliver all orders by the least expensive, and thereby slower, mode, resulting in excessively late orders. This penalty value can be altered depending on the importance the company places with on-time deliveries. Some companies have policies in which late deliveries are unacceptable, while others will allow it in those cases where the extra profits justify the delay.

2.3. Business Policies

We next define two important policies used in our modeled business scenario.

2.3.1. Service Levels

A major area of exploration involves the concept of service levels. When an order is placed, the customer chooses a desired *service level*, rather than

transportation mode. This service level corresponds to the lead time from the time the order is placed until the time it *arrives* to the customer. With this policy, the manufacturing company can determine when to ship the order and by what mode to optimize its production schedule, and ultimately, its revenue.

This policy contrasts the typical policy in place at most companies today, in which the customer knows the date the product will be ready for shipment, and simply chooses the transportation mode (henceforth referred to as the ‘standard policy’). With this policy, the company has limited flexibility in the production schedule because the order must be ready to ship by the set date. Additionally, it has no flexibility in choosing the transportation mode; it simply ships by the customer-requested mode.

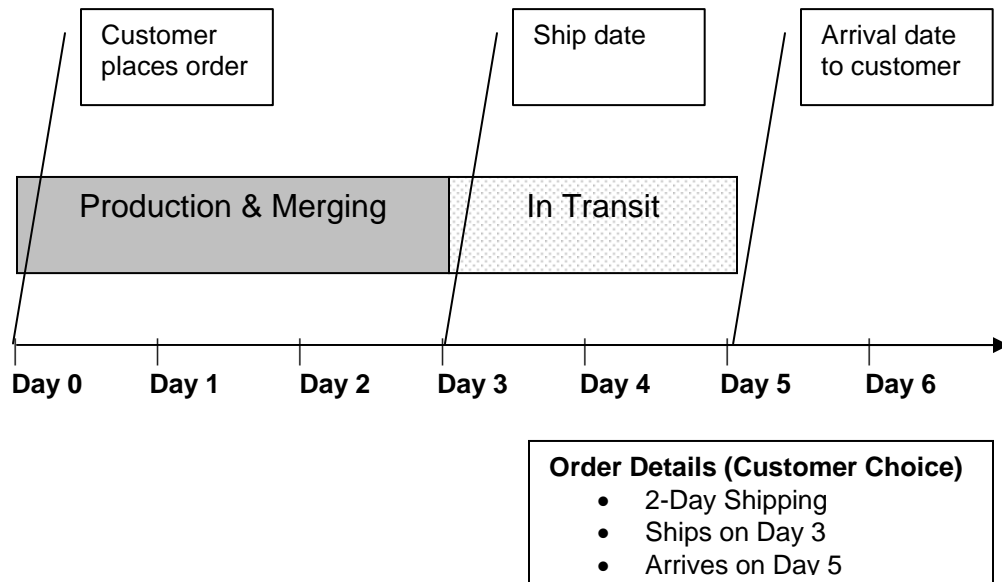
In our business scenario, the user is able to set the number of service levels available to the customer. For example, the company can offer three service levels – Gold, Silver or Bronze. Each service level then has an associated lead time (the days from order placement until arrival at the customer). The premium service levels will have shorter lead times, but will have higher associated costs. Thus, the customer must trade off the extra costs for faster delivery in choosing the desired service level.

The model tests this new policy to determine its effect on profits and commitment levels. For instance, with the standard policy a customer can select 2-Day delivery, at a predetermined cost (given a production lead time of three days for an arrival date five days out). With the new policy, the comparable service

level for the customer might be Silver, which corresponds to a five day lead time. The company could then opt to produce and ship the order by any mode and date, as long as the order arrives on the fifth day.

With both policies, the customer would receive the order on the specified day, five days from when the order was placed. However, the interesting aspect of the new policy is that the company can choose to produce the order later than normal and ship via 1-Day shipping, if perhaps a higher profit margin order comes in with immediate production needs. Due to other demand, it might be advantageous to delay production and ship by a faster mode. The added shipping costs would be offset by the additional revenue gained from fulfilling additional demand with higher profit margins. The following diagram illustrates this new service level concept.

Standard policy: customer chooses mode for order delivery



Our model policy: customer chooses service level for order

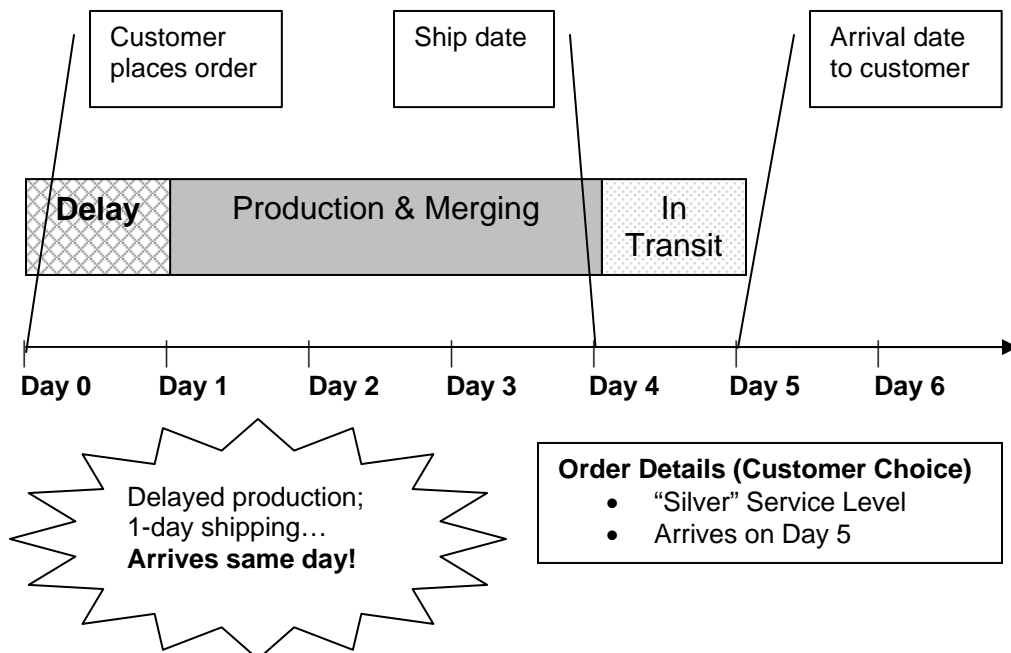


Figure 2.2: Service Level Overview

2.3.2. Commitment

In our business case, the accepted orders are input to the model. In most cases, this would correspond to data from the order processing system. These orders have already been accepted and must be delivered to the customer. However, our model might schedule orders to be delivered late for a due date violation penalty. This would occur in the case of limited capacity and resources, or in the case when the production is delayed to fulfill higher profit margin demand orders.

Each order can be split and delivered in separate shipments to the customer, as long as the deliveries all arrive on the same day. The number of acceptable splits must be specified in the data input to the model.

A major part of the model is in determining the commitment of forecast demand. The model decides when resources (production capacity and materials) should be reserved for future demand. When beneficial, the model will accept and reserve capacity for future demand, especially in the case of high profit configurations. This lead-time setting enables the company to dedicate a portion of capacity to the more valuable future orders.

Because the values for demand are aggregate figures, the commitment level is modeled as a percentage. Thus, a demand can be 40% accepted, for example. This should not be misconstrued as a policy that partial orders are accepted (e.g. only a certain quantity of the full requested amount is fulfilled). Rather, because the values are aggregate, it merely represents the case that some future orders are not committed, while some are committed fully. This policy is tested later in one

of the experiments of the model. The model does not allow for future demand to be committed if it cannot be delivered on-time. Due date violations are only associated with orders.

2.4. Assumptions

In the course of translating the business scenario into the model, a number of assumptions were made. It was necessary to trade-off simplification of certain areas to create a model that could run efficiently. Certainly, the model could be enhanced to include additional features if desired.

When considering production, the model determines the production schedule for each factory, but does not further specify a particular assembly line or exact hour for production. It is assumed that each factory uses a more powerful scheduling system which could take into account the nuances of the factory, such as available workers, machine down-time, etc. We simply input an overall daily production capacity for the factory in our model. Finally, we ignore the production costs for assembly of each product. We assume that each factory has the same cost to build each product. Since we are dealing with the profit margins of each product when calculating the revenue, the manufacturing costs have already been accounted for.

We ignore the inventory costs to store parts, Kitting SKUs, and Merging SKUs at the factories and merging centers. As the model tries to minimize the overall timeframe between order placement and order arrival, few products will be

stored in inventory for extended amounts of time. We also assume that the transportation modes do not have associated lot size minimums or maximums.

Our model is not a rolling time model; it only considers orders and demand for the given day. So, orders from the previous day are not re-evaluated as far as production scheduling is concerned. Additionally, forecast demand for two days in the future is not considered when reserving capacity. The values provided to the model for resources should reflect this, and should correspond to the *fraction* of capacity available on each particular day for new orders/demand.

An additional assumption is in the treatment of forecast demand that resources are not reserved for. In practice, if demand is not reserved at a certain product and service level configuration, a portion of that demand would shift to another service level. For instance, if a forecast order for a product at the Gold service level cannot be reserved, a high fraction of that demand would then move to the next available service level. Our model does not account for this demand – in this sense, we assume that this demand is lost entirely if it cannot be committed.

Finally, the model uses simplified pricing schemes for profit margins and transportation costs. The formulation of these terms is discussed later in the section on data creation. Additionally, the demand forecasting module is fairly simplified; the purpose of the model is not to accurately predict demand levels, but rather to analyze resources *given* a demand.

2.5. Model Formulation

The problem was formulated as a mixed integer program. The formulation, including all given parameters, decision variables, objective function and constraints is detailed below:

Parameters

Indices

t = Time periods in model horizon

k = Customer order

d = Forecast demand

s = Service level

l = Transportation mode

i = SKUs of kitting and merging

j = Kitting parts (for assembly of Kitting SKUs at factory)

f = Factories

m = Merging centers

Orders

y = Number of times an order can be split during delivery

ao = Average quantity of SKUs per order

oq^k = Quantity of order k

os^k = Service level for order k

$oc^{i,k}$ = Configuration for order k (quantity of SKU i needed)

$ol^{m,k}$ = Merging center m affiliated with order k

Forecast Demand

dq^d = Quantity of demand d

ds^d = Service level for demand d

$dc^{i,d}$ = Configuration for demand d (quantity of SKU i needed)

$dl^{m,d}$ = Merging center m affiliated with demand d

Costs

w' = Weight given for profits in objective function

w'' = Weight given for costs in objective function

w''' = Weight given for due date violation in objective function

u^l = Fixed costs for transportation mode l

v^l = Variable costs for transportation mode l

$x^{i,s}$ = Profit margin for SKU i at service level s

Production/Merging/Delivery

$b^{i,j}$ = Bill of material of parts j for production of Kitting SKUs i

$p^{f,t}$ = Total production capacity for factory f and time period t

$sf^{f,m}$ = Lead time to transfer SKUs from factory f to merging center m

sm^l = Delivery lead time for transportation mode l from merging center m

sl^s = Required time for delivery for service level s

Inventory

$pj^{f,j,t}$ = Incoming supply of kitting part j at factory f and time period t

$yj^{f,j}$ = Initial inventory of kitting part j at factory f

$yf^{f,i}$ = Initial inventory of Kitting SKU i at factory f

$yk^{m,i}$ = Initial inventory of Kitting SKU i at merging center m

$ym^{m,i}$ = Initial inventory of Merging SKU i at merging center m

$pm^{m,i,t}$ = Incoming supply of Merging SKU i at merging center m and time period t

Decision Variables

Orders/Demand

D^d = Resource reservation status for demand d : (%)

$LO^{k,l,t}$ = Delivery status for order k by transportation mode l in time period t ; *binary*

$LD^{d,l,t}$ = Delivery status for demand d by transportation mode l in time period t ; *binary*

$QO^{k,l,t}$ = Delivery quantity for order k by transportation mode l in time period t

$QD^{d,l,t}$ = Delivery quantity for demand d by transportation mode l in time period t

$AO^{k,l,t}$ = Arrival quantity for order k by transportation mode l in time period t

Costs/Profits

H^k = Profit from order k

E^d = Profit from reserved demand d

CO^k = Cost of delivery for order k

CD^d = Cost of delivery for reserved demand d

DD = Due date violation penalty costs

Production/Inventory

$Z^{f,i,t}$ = Total production quantity at factory f of Kitting SKU i during time period t

$N^{f,m,i,t}$ = Quantity of Kitting SKU i transferred from factory f to merging center m per time period t

$ZJ^{f,j,t}$ = Inventory level of part j at factory f in time period t

$ZM^{m,i,t}$ = Inventory level of of Merging SKU i at merging center m in time period t

$ZK^{m,i,t}$ = Inventory level of Kitting SKU i at merging center m in time period t

$ZF^{f,i,t}$ = Inventory level of Kitting SKU i at factory f in time period t

Objective Function :

Maximize Profit

$$w^1 \cdot \sum_{k \in K} H^k + (1 - w^1) \cdot \sum_{s \in S} E^d - w^2 \cdot \left(\sum_{k \in K} CO^k + \sum_{d \in D} CD^d \right) - w^3 \cdot \sum_{k \in K} DD^k$$

Total profits are dependent on profits from committed orders and demand, less the associated delivery costs and the due date violation costs

Subject to:

(1) *Profits and Costs Definition*

$$(1.1) \quad H^k = \sum_{i \in I, l \in L, t \in T} oc^{i,k} \cdot x^{i,os^k} \cdot QO^{k,l,t} \text{ for all } k \in K$$

Order profits are dependent on the particular configuration of the order, the profit margin and the quantity of the order that was delivered during the model timeframe

$$(1.2) \quad E^d = \sum_{i \in I} dc^{i,d} \cdot x^{i,ds^d} \cdot dq^d \cdot D^d \text{ for all } d \in D$$

Demand profits are dependent on the particular configuration of the demand, the profit margin and quantity reserved

$$(1.3) \quad CO^k = \sum_{l \in L, t \in T} \left(u^l \cdot (1/ao) \cdot QO^{k,l,t} + v^l \cdot QO^{k,l,t} \right) \text{ for all } k \in K$$

Order delivery costs are based on the fixed costs of each committed order as well as the variable costs related to order quantity

$$(1.4) \quad CD^d = \sum_{l \in L, t \in T} \left(u^l \cdot (1/ao) \cdot QD^{d,l,t} + v^l \cdot QD^{d,l,t} \right) \text{ for all } d \in D$$

Demand delivery costs are based on the fixed costs as well as the variable costs related to each delivery quantity

$$(1.5) \quad DD^k = \sum_{t=sl^{os^k}+1}^t \left(\left(t - sl^{os^k} \right) \cdot \sum_{l \in L} AO^{k,l,t} \right) + \left(t + \text{Max}(sm^l) - sl^{os^k} \right) \cdot \left(oq^k - \sum_{l \in L, t \in T} AO^{k,l,t} \right) \text{ for all } k \in K$$

Due date violation is the number of days past the requested service level due date that that an order arrives, including order quantities that are not delivered at all within the model timeframe

(2) *Order Delivery*

$$(2.1) \quad \sum_{l \in L, t \in T} LO^{k,l,t} \leq y \text{ for all } k \in K$$

Orders must be delivered within allowable number of delivery splits

$$(2.2) \quad QO^{k,l,t} \leq oq^k \cdot LO^{k,l,t} \text{ for all } k \in K, l \in L, t \in T$$

The delivery quantity each day (by each method) must be less than the requested amount

$$(2.3) \quad LO^{k,l,t} \leq QO^{k,l,t} \text{ for all } k \in K, l \in L, t \in T$$

An order status is considered delivered by a certain transportation method only if an actual quantity is delivered to the customer

$$(2.4) \quad \sum_{l \in L, t \in T} QO^{k,l,t} \leq oq^k \text{ for all } k \in K$$

The total amount delivered cannot be more than the requested amount

$$(2.5) \quad QO^{k,l,t} = AO^{k,l,t+sm^l} \text{ for all } k \in K, l \in L, t \in T$$

The arrival of an order is dependent on the ship date of the order plus the delivery lead time

$$(2.6) \quad AO^{k,l,t} = 0 \text{ for all } k \in K, l \in L, t \in T \mid t \leq sm^l$$

The arrival of an order cannot occur in the beginning of the model, within the lead time for the specified delivery method

(3) *Demand Delivery*

$$(3.1) \quad \sum_{l \in L, t \in T} LD^{d,l,t} \leq 1 \text{ for all } d \in D$$

Demand must be delivered in the same shipment (no splits)

$$(3.2) \quad \sum_{l \in L, t \in T} (t + sm^l) \cdot LD^{d,l,t} \leq sl^d \text{ for all } d \in D$$

Demand must be delivered within allowable service level date

$$(3.3) \quad QD^{d,l,t} \leq dq^d \cdot LD^{d,l,t} \text{ for all } d \in D, l \in L, t \in T$$

The delivery quantity cannot be more than the requested amount

$$(3.4) \quad D^d \leq \sum_{l \in L, t \in T} LD^{d,l,t} \text{ for all } d \in D$$

Demand must be delivered if resources are reserved

$$(3.5) \quad \sum_{l \in L, t \in T} QD^{d,l,t} = dq^d \cdot D^d \text{ for all } d \in D$$

When resources are reserved for demand, the quantity delivered must equal the percent reserved of demand

(4) *Material Conservation*

$$(4.1) \quad ZJ^{f,j,t=0} = yj^{f,j} \text{ for all } f \in F, j \in J$$

Initial inventory of kitting parts at each factory

$$(4.2) \quad ZJ^{f,j,t} = ZJ^{f,j,t-1} + pj^{f,j,t} - \sum_{i \in KI} b^{i,j} \cdot Z^{f,i,t} \text{ for all } f \in F, j \in J, t \in T$$

Daily inventory of kitting parts is the previous day's inventory combined with stock supply, less parts used for production

$$(4.3) \quad \sum_{i \in KI} Z^{f,i,t} \leq p^{f,t} \text{ for all } f \in F, t \in T$$

Production must be within capacity for each factory

$$(4.4) \quad ZF^{f,i,t=0} = yf^{f,i} \text{ for all } f \in F, i \in KI$$

Initial inventory of Kitting SKUs at each factory

$$(4.5) \quad ZF^{f,i,t} = ZF^{f,i,t-1} + Z^{f,i,t} - \sum_{m \in M} N^{f,m,i,t} \text{ for all } f \in F, i \in KI, t \in T$$

Inventory of Kitting SKUs at each factory is the previous day's inventory plus the quantity produced, less the quantity transferred to merging centers

$$(4.6) \quad ZK^{m,i,t=0} = yk^{m,i} \text{ for all } m \in M, i \in KI$$

Initial inventory of Kitting SKUs at each merging center

$$(4.7) \quad ZK^{m,i,t} = ZK^{m,i,t-1} + \sum_{f \in F | flt^{f,m} < t} N^{f,m,i,t-sf^{f,m}} - \sum_{l \in L, k \in K | ol^{m,k}=1} oc^{i,k} \cdot QO^{k,l,t} \\ - \sum_{l \in L, d \in D | dl^{m,d}=1} dc^{i,d} \cdot QD^{d,l,t} \text{ for all } m \in M, i \in KI, t \in T$$

Inventory of Kitting SKUs at each merging center is the previous day's inventory plus the quantity transferred in from each factory (accounting for the transfer lead time), less the quantity shipped to customers

$$(4.8) \quad ZM^{m,i,t=0} = ym^{m,i} \text{ for all } m \in M, i \in MI$$

Initial inventory of Merging SKUs at each merging center

$$(4.9) \quad ZM^{m,i,t} = ZM^{m,i,t-1} + pm^{m,i,t} - \sum_{l \in L, k \in K | ol^{i,k}=1} oc^{i,k} \cdot QO^{k,l,t} \\ - \sum_{l \in L, d \in D | dl^{i,d}=1} dc^{i,d} \cdot QD^{d,l,t} \text{ for all } m \in M, i \in MI, t \in T$$

Inventory of Merging SKUs at merging centers is the previous day's inventory combined with daily stock supply, less amount shipped for orders and demand

3. Model Implementation

This chapter provides information as to how the model was implemented. We describe the system architecture, including the use of Excel and Xpress for the model. We then cover the aspect of data formulation and creation. Finally, we conclude with a discussion of the specifications of the model, as far as computational time and size is concerned.

3.1. System Architecture Design

When designing the system architecture for the model implementation, we wanted to balance two contradictory goals – selecting optimization software powerful enough to solve the MIP with thousands of variables, and choosing a simple, flexible setup designed for our target user. We intended for the main users of the model to be business managers, production schedulers, sales groups, etc., who may not necessarily be versed in technical programming. Consequently, we wanted a system that would be intuitive for users to learn quickly, yet could handle the complexity of the model.

We decided to implement the model using a combination of Xpress-MP callable solver and Microsoft Excel. This combination results in maximum ease of understanding and flexibility for the end users, while still maintaining the strength of the model. The front-end of the model is through Excel and the back-end processing is done by Xpress. While perhaps lesser known than CPLEX, Xpress is

gaining use for production scheduling, logistics and e-commerce applications. This optimization package is equipped to solve extremely large MIP models within reasonable computational time. Additionally, it can easily connect to Excel to transfer data and results. In fact, once we formulate the model in Xpress, users can call and run the model completely within Excel. Thus, the model can be used easily and even modified by someone with little to no programming or formulation experience.

To set up the model, we first translated the mathematical model formulation into Mosel code in Xpress. This file is compiled and stored as a binary model (BIM). Within Excel, we set up data tables to store the input to the model (order and demand details, plus parameters like transportation costs, BOMs, production capacities, etc.). When the user initiates the model in Excel, a VB macro (which uses the Xpress-MP-callable libraries) runs the model using Xpress, and retrieves the results for analysis in Excel. The following diagram shows the technical setup of the model.

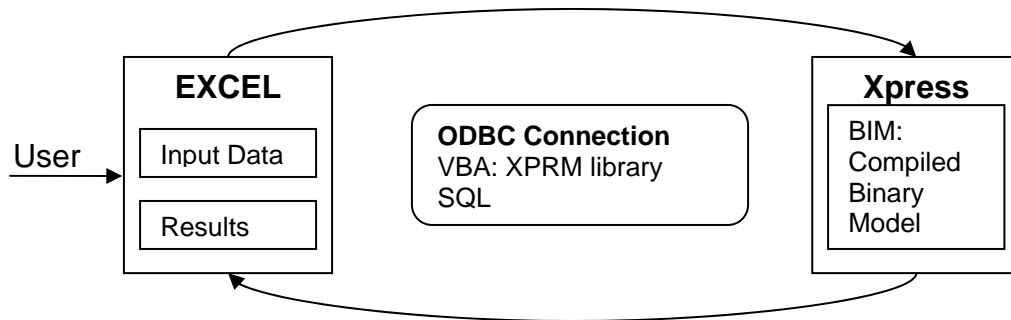


Figure 3.1: High-level System Architecture

3.2. Software Implementation

3.2.1. Use of Xpress

Xpress-MP is a commercial software package developed by Dash Optimization, and was chosen due to its ability to efficiently handle the integer program and the high volume of decision variables. Xpress-MP is a suite of optimization tools that include optimizer algorithms, the IVE visual development environment and Mosel, a modeling and optimization environment and language.

The optimizer algorithms include simplex (both primal and dual), the Newton barrier optimizer, and a branch-and-bound framework used for mixed integer programming problems (MIP). The MIP optimizer was used to solve our model. It uses a sophisticated branch-and-bound algorithm to quickly identify solutions; the cutting plane strategies involve flow covers, GUB covers, lift and project, cliques, flow paths, and Gomory fractional cuts. The MIP presolve algorithm preprocesses the problem to reduce the size and to cut down on the final solving time. Searches can be customized for breadth-first, depth-first or best-first.

Xpress-IVE is an integrated modeling and optimization development environment for Windows. It incorporates the Mosel program editor, compiler, and execution environment.

Mosel is the programming language used within Xpress. It was created to be as close to the algebraic formulation as possible, which leads to generally

understandable code. Our Mosel code of the model formulation is provided in Appendix A.

Once we formulated the model in Mosel, we compiled it into a BIM file. When the model is executed, this file is passed to the MIP Optimizer to solve. The following diagram gives the setup within Xpress

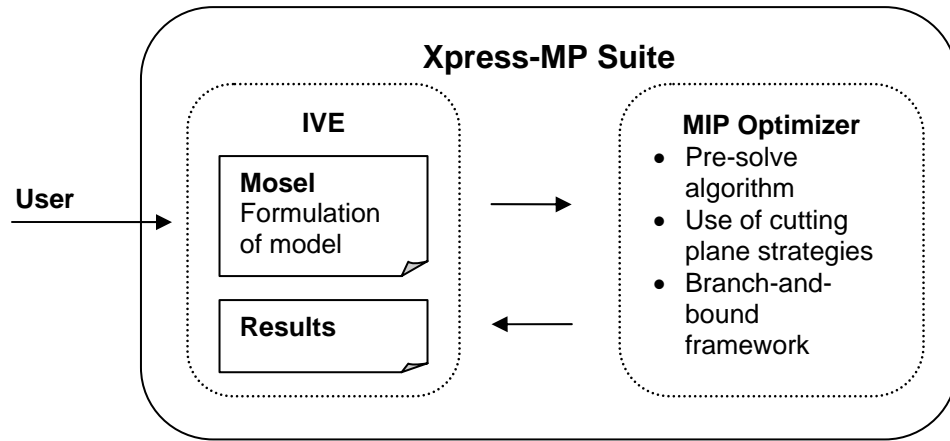


Figure 3.2: Details of Xpress-MP Suite

3.2.2. Use of Excel

We chose to use Excel for data management and results analysis for our model. Undoubtedly, we could have chosen a more powerful database tool. However, the data relationships in Excel are more transparent to the user; plus, the data is formatted and displayed for quick updates and analysis. Additionally, the data structure for our model is not so complex as to warrant the use of Oracle, Access or another database system.

Obviously, one major drawback with using Excel is the limitation on model size. However, in our trial runs of the model we were able to store the necessary

data without any loss of clarity and without any computational issues within Excel. Clearly, if a user intends to use the model for actual day-to-day production setting and order promising, a more robust database would be needed. However, for the purposes of analyzing general trends and testing policies, Excel is more than sufficient.

The input data for the model is thus stored in tables within an Excel spreadsheet. The user must provide the initial parameters to define the model scenario, as well as provide data for the orders and demand. First, the index parameters must be specified. These indexes specify the identifiers for the other model data parameters. Additionally, they are the indexes for the decision variables of the model. The tables identify the valid entries for each index. These entries are of *string* format. For instance, for service levels, the valid entries could be Gold, Silver and Bronze. The following table details the model indexes.

Indexes
Factories
Merging Centers
Kitting Parts
Kitting SKUs
Merging SKUs
Transportation Modes
Service Levels
Time Periods

Table 3.1: Model Indexes

Next, the user must specify the parameter values. These tables contain all the data setup values for the model. These can be changed from one run to the next run to test different scenarios. And, because the tables are in Excel, the values can

be derived easily from formulas. For instance, transportation costs are based on a formula for pricing, calculated from information in a separate spreadsheet tab. For each of these tables, the subsequent data is of type *real*. As an example, the initial inventory of kSKU1 = 20 at Factory A, 30 at Factory B, etc. The following table specifies the parameter tables in Excel.

Parameter	Uses Index(es)
Initial inventory of kSKUs at factories	kSKUs, Factories
BOMs	kSKUs, Parts
Initial inventory of parts	Parts, Factories
Part stock	Parts, Factories, Time
mSKU stock	mSKUs, Merging Centers, Time
Lead time from factories to merging centers	Factories, Merging Centers
Initial inventory of mSKUs	mSKUs, Merging Centers
Service Level lead time	Service Levels
Transportation mode lead time from Merging Centers to customer	Transportation modes
Transportation mode fixed costs	Transportation modes
Transportation mode variable costs	Transportation modes
Profit margins	SKUs, Service Levels

Table 3.2: Parameters and Associated Indexes

Finally, the user must provide information regarding orders and demand.

The following diagram specifies the required data.

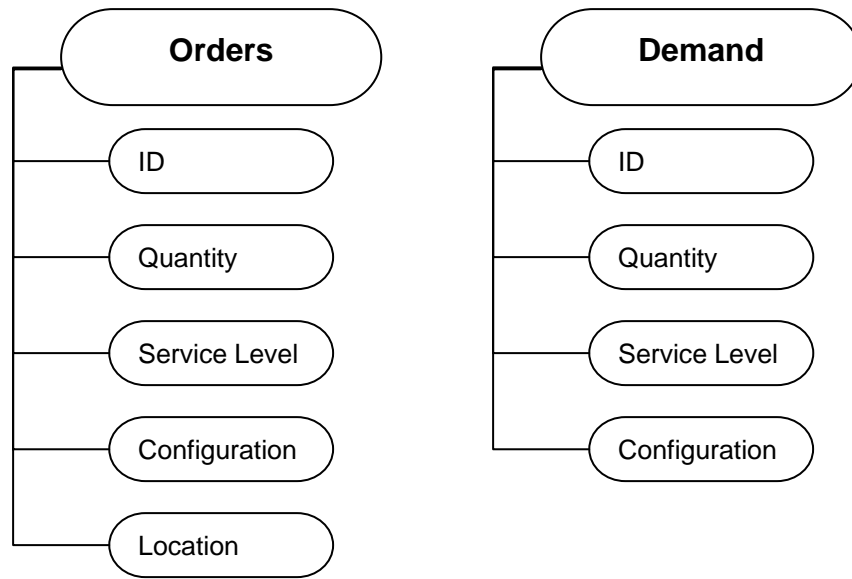


Figure 3.3: Required Data for Orders and Demand

Excel was also used to present the results of the model to the user. The data is presented in simple tables, detailing the most important decision variables (demand commitment percent, shipping schedule, production schedules, etc.). Using the analysis functionality of Excel, the user can summarize quickly the results of the model and analyze the trends.

3.2.3. Interaction of Xpress and Excel

As we have mentioned, once the formulation has been generated in Xpress, the model can be run entirely from Excel. To facilitate this, a connection must be established so that data can be passed back and forth effectively. In Excel, we programmed a module using Visual Basic for Applications (VBA) that would call the Xpress solver to import and solve the model. We added the appropriate Xpress module to the VB project (XPRM) and added the library xprmb.dll to the correct

directory. This enabled the Mosel VB interface to allow the Mosel runtime and compiler libraries to be called from within VBA code.

The VB code loads the BIM file (compiled model formulation) into Mosel and then executes the model. The results are then pulled back into preset tables in Excel using VB scripting. The elements of each requested decision variable can be retrieved individually, or can be summed or otherwise manipulated for analysis in Excel. Finally, a log file is also generated to detail any issues during execution. A portion of the VB code used to run the model is included in Appendix B.

On the Xpress side, commands must be added to the Mosel formulation to establish the connection. First, we marked the Excel workbook as a Data Source (DSN) within our computer's ODBC settings. Next, we added the ODBC I/O driver (mmodbc) to the Mosel code to allow access to external data sources. The input data values are accessed by Xpress through a series of SQL statements. Within Excel, we defined each data table as a *named range*. These data ranges are then pulled and used to fill the associated data arrays in the BIM. The following diagram presents an overview of the interaction.

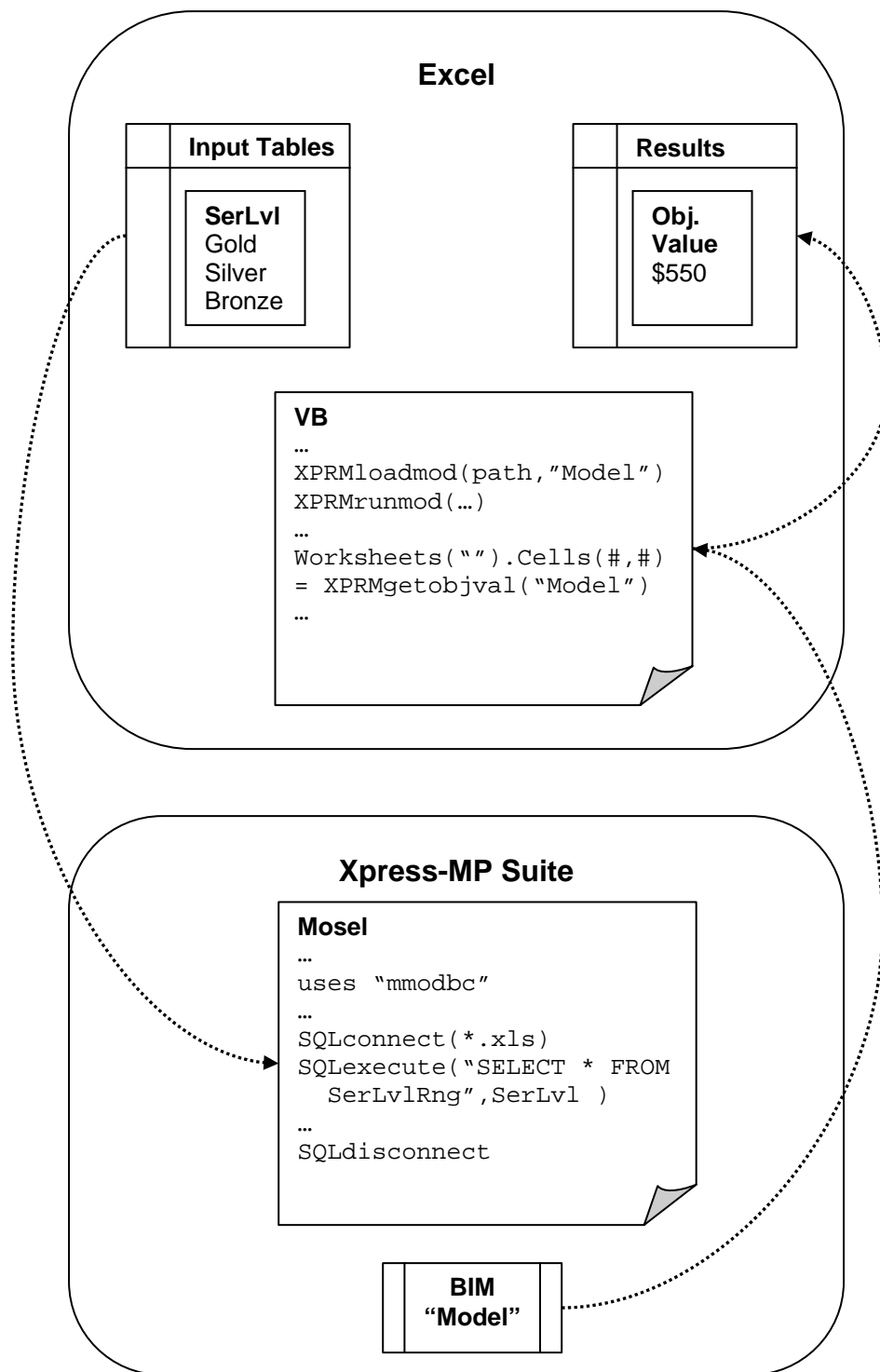


Figure 3.4: Interaction of Excel and Xpress

3.3. Data Setup

The following sections detail the data setup for the model. We first discuss the selected size of the model (number of indexes and orders/demand). Then, we describe how data was formulated and generated for our trial runs.

3.3.1. Model Size

For our analysis of the model, we wanted a model that was large enough to provide sufficient results, yet not so large as to become overburdened in computational time. When we were setting the parameter size, we were careful to keep the size in check. We chose to analyze two factories and three merging centers. Thus, we could study differences resulting in production shortages at one factory to see how production shifts. It was important to have one more merging center than factory to analyze how the model divided finished products amongst the merging centers.

We chose to represent five product families in our base setup - three Kitting SKU product families and two Merging SKU product families. The Kitting SKUs are assembled from an array of 10 parts. Some of the parts are shared, while some are unique to each Kitting SKU. This variety enabled testing on the differences due to profit margins and shared resources between the product families.

Additionally, the orders could be shipped to the customer by one of three transportation modes. A rush mode was setup, with the highest cost, as well as two slower modes with corresponding costs. Finally, we chose to have three service

levels, which would provide adequate differences in expected delivery dates and profit margins.

The number of incoming orders and forecast demand is based on the number of Merging SKUs, Kitting SKUs, merging centers and service levels. For our analysis purposes, each order/demand has a configuration of a single product. Each order is designated a merging center, based on the geographic proximity to the customer. Although forecast demand is not specified for a geographic region, we estimate the fraction served by each merging center to generate an assignment. Additionally, each order/demand has an associated Service Level. So, to analyze all the various combinations, we need 45 orders and 45 demands ($3 * 3 * (2+3)$). The following table provides details of the model size.

Parameters	Dataset Size
Factories	2
Merging Centers	3
Service Levels	3
Kitting SKUs	3
Kitting Parts	10
Merging SKUs	2
Transportation Modes	3
Orders	45
Demand	45

Table 3.3: Dataset Size of Parameters

3.3.2. Data Generation for Experiments

Unfortunately, we did not have any real production data to use during our model runs. However, we generated data that mimicked actual data so that our analysis and conclusions would be accurate.

Perhaps the most complex aspect of data creation was developing the pricing scheme for shipping costs and profit margins. Each shipping mode has its own related costs. For each mode, there is a fixed cost associated with the shipment and a variable cost based on the size of the order. Clearly, the actual shipping cost for a product is dependent on size, weight and exact distance between the customer and the merging center. We simplified the pricing considerably so as not to overly complicate the model. We assumed each product was roughly the same size/weight. For our model analysis, we assumed the products were computers, and then analyzed the posted prices from a website of a leading computer company to estimate the shipping prices. We collected a set of data for similarly sized/priced products at different quantities for each of the corresponding transportation modes (1-Day Air, 2-Day Air, and 3-Day Ground). Next, we performed regression analysis to determine a simplified formula that could be used for our model. The first term is for the fixed costs, and the second term is added in for each additional quantity in the shipment. The resulting formulas are shown in the following table.

Transportation Mode	Shipping Costs Formula
1-Day Air	$135.99 + 167.00 * (Qty - 1)$
2-Day Air	$110.00 + 123.25 * (Qty - 1)$
3-Day Ground	$81.33 + 103.42 * (Qty - 1)$

Table 3.4: Formulation of Shipping Costs per Transportation Mode

These formulas represent the shipping *price* charged for each shipping mode. However, we assumed that the company marked up the actual shipping costs by a percentage to increase revenue. We needed the actual *cost* the company

incurs for shipping the product for our model. Hence, we reduced these values by a margin to get the revised formulas used in the model, which are presented in the following table.

Transportation Mode	Revised Shipping Costs Formula
1-Day Air	$90.66 + 111.34 * (Qty - 1)$
2-Day Air	$74.00 + 82.17 * (Qty - 1)$
3-Day Ground	$54.22 + 68.94 * (Qty - 1)$

Table 3.5: Revised Formulation of Shipping Costs per Transportation Mode

The following graph provides an overview of the actual costs per transportation mode, across varying quantities. Our model assumed the average shipment size is 10 products (based on data from a leading computer manufacturer), so the aggregate order and demand values are divided into shipments of 10 products, or fraction thereof, for delivery to the customer.

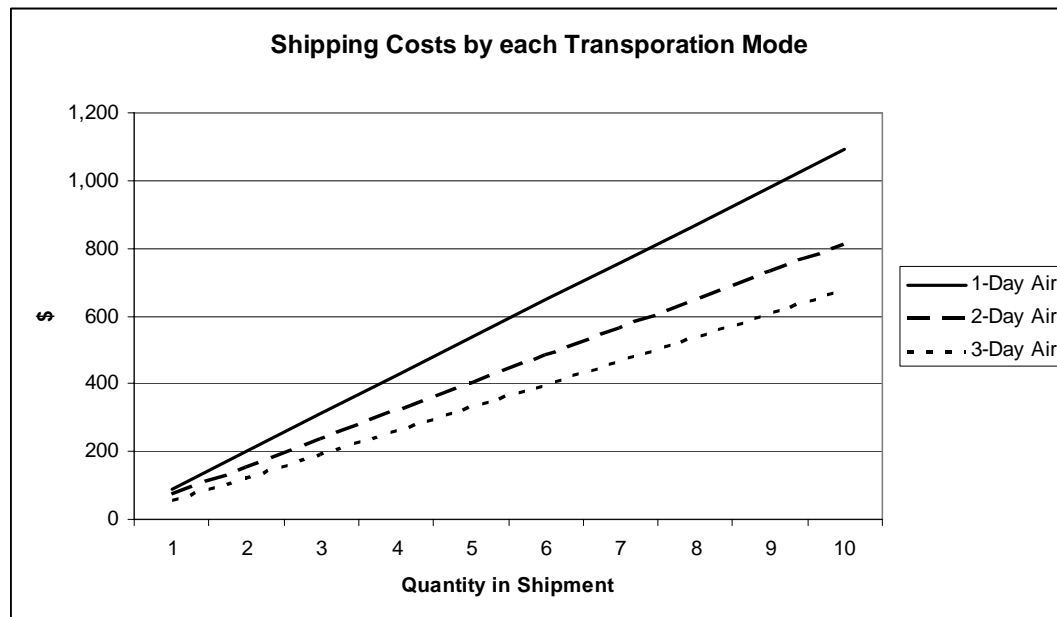


Figure 3.5: Graph of Shipping Costs per Transportation Mode

Our next task was to create formulas to use for the pricing of each product at the various service levels. We needed a formula to generate the profit margin of each service level for the different SKUs. We assigned approximate *list prices* for each of our products, which would correspond to the sales price the customer would pay for the product. This price does not include shipping. These prices were set so that the average was roughly \$1,000, with some variety in the prices to differentiate the products. Now, for our model, we needed to translate the *list price* to the *cost* of the product to determine the *profit margin* for each product. The first step was to determine the cost of the product before the markup to the customer. We assumed a markup of 25%.

Our model differs from the standard policy at most companies in that the customer selects the service level for the product and pays a combined list price, which includes both the product and the shipping. When developing our pricing model, we needed to account for this. Essentially, the profit margin for a given product and service level combines the product profit margin and the shipping profit margin. Recall earlier that we assume the customer is also charged a markup on the shipping costs.

We associated each service level with a comparable shipping mode to determine the corresponding shipping profit margins. Therefore, instead of choosing 1-Day Air as the shipping mode, the customer would choose the ‘Gold’ service level, and so on for each of the service levels and modes. Thus, we equate the profit margin on 1-Day Air to the Gold service level in the pricing scheme.

Finally, we assume that the company can add an extra premium for the higher service levels. We set these premiums at 5% for Gold, 2% for Silver, and 0% for Bronze. The following table details how the profit margins were set.

		kSKU1	kSKU2	kSKU3	mSKU1	mSKU2
	Standard sales price (does not include shipping)	999.00	1,539.00	799.00	750.00	849.00
G O L D	Product markup	25%	25%	25%	25%	25%
	Product profit	199.80	307.80	159.80	150.00	169.80
	Service Level markup	5%	5%	5%	5%	5%
	Service Level profit	239.76	369.36	191.76	180.00	203.76
	1-Day Air shipping cost	90.66	90.66	90.66	90.66	90.66
	Shipping markup	50%	50%	50%	50%	50%
	Shipping profit	45.33	45.33	45.33	45.33	45.33
	Total Profit Margin	285.09	414.69	237.09	225.33	249.09
	Gold Service Level sales price	1,174.95	1,736.55	966.95	915.99	1,018.95
	Standard sales price, incl. 1-Day Air shipping	1,134.99	1,674.99	934.99	885.99	984.99
S I L V E R	Product markup	25%	25%	25%	25%	25%
	Product profit	199.80	307.80	159.80	150.00	169.80
	Service Level markup	2%	2%	2%	2%	2%
	Service Level profit	15.98	24.62	12.78	12.00	13.58
	2-Day Air shipping cost	74.00	74.00	74.00	74.00	74.00
	Shipping markup	50%	50%	50%	50%	50%
	Shipping profit	37.00	37.00	37.00	37.00	37.00
	Total Profit Margin	252.78	369.42	209.58	199.00	220.38
	Silver Service Level sales price	1,125.98	1,674.62	922.78	873.00	973.58
	Standard sales price, incl. 2-Day Air shipping	1,110.00	1,650.00	910.00	861.00	960.00
B R O N Z E	Product markup	25%	25%	25%	25%	25%
	Product profit	199.80	307.80	159.80	150.00	169.80
	Service Level markup	0%	0%	0%	0%	0%
	Service Level profit	0.00	0.00	0.00	0.00	0.00
	3-Day shipping cost	54.22	54.22	54.22	54.22	54.22
	Shipping markup	50%	50%	50%	50%	50%
	Shipping profit	27.11	27.11	27.11	27.11	27.11
	Total Profit Margin	226.91	334.91	186.91	177.11	196.91
	Bronze Service Level sales price	1,080.33	1,620.33	880.33	831.33	930.33
	Standard sales price, incl. 3-Day shipping	1,080.33	1,620.33	880.33	831.33	930.33

Table 3.6: Formulation of Profit Margins

The following chart breaks down the profit margins for each service level and SKU. We can see that Kitting SKU2 has the highest profit margins, followed by Kitting SKU1.

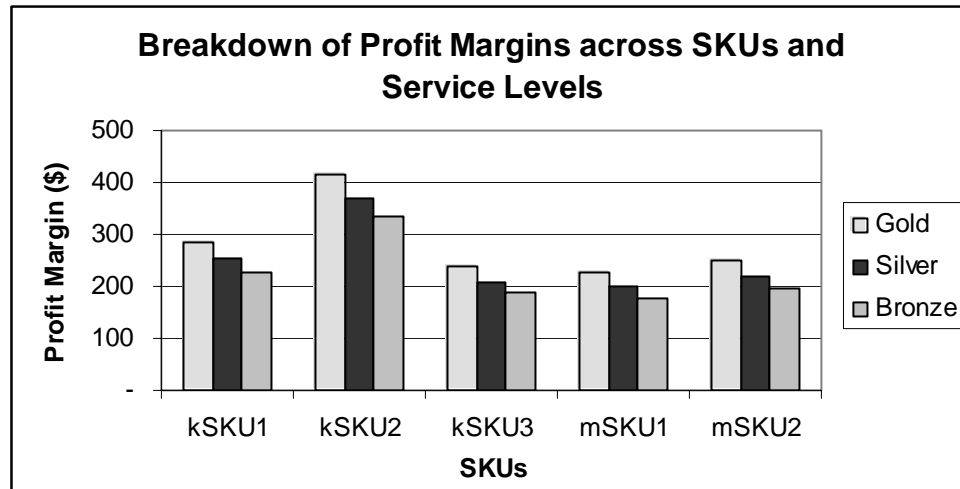


Figure 3.6: Chart of Profit Margins per SKU and Service Level

We next describe the final parameter values that we set for the model, including initial inventory levels and various lead times. The inventory levels were set arbitrarily, but mainly reflected the fact that limited inventory would be carried from day to day. We next set the lead time for production and shipping from each factory to each merging center. We simplified the transfer system and assumed that if the factory and merging center are located at the same site, the lead time is one day, which includes the time to assemble the product and the time to transfer to the merging center. If the factory and merging center are not co-located, then the lead time is set at two days. Finally, we choose the service level lead times. The minimum production/transfer to merging center is one day. Furthermore, the fastest delivery method is 1-Day Air, so the quickest arrival date to the customer

would be on Day 3 (Day 1: production/transfer, Day 2: ship from merging center, Day 3: arrive at customer). So, we set the Gold service level (premium) at 3 days, Silver at 5 days and Bronze at 7 days.

3.3.3. Generation of Data

To test each business scenario of the model, several trials must be executed for each run. By having the values for certain parameters fluctuate from trial to trial, we can see how the model acts generally in each scenario. To facilitate this, we set up a program in Visual Basic to randomize certain parameters.

First, we had to decide which factors to randomize, and which to keep at constant values throughout all trials. We kept the model size the same, so that subsequent comparable analysis could be performed (e.g.: we did not vary the number of factories which would create a profound effect on the results of the model). We chose to vary the following parameter values:

- Order quantity
- Demand quantity
- Production capacity at the factories
- Incoming supply of kitting parts at the factories
- Incoming supply of Merging SKUs at the merging centers

For each parameter, the values are randomized so that they are uniformly distributed over a range. In the general business scenario we are modeling, the lump sum quantity of requested products for demand and orders is roughly equivalent to the capacity needed to fulfill all orders and demand. Thus, to capture this setting, we made certain parameters dependent on the value of other parameters.

We started with the order quantity, an independent value. We set the range at 550 to 700 units per order of each product and service level. This value would then be divided among the orders of that product/service level for each of the three merging center assignments, depending on the set distribution schedule. Next, we looked at the demand quantity. The purpose of this model is not to predict demand, and as such, we used a simple setting that assumed the quantity for incoming demand would be roughly the same as the known orders. We expanded the range slightly, and randomized the demand quantity between 500 and 800 units per demand of a certain product and service level. Like the order quantity, this would then be distributed among demand forecasts for the three merging centers.

We next considered the production capacity values. We wanted capacity to be approximately equal to the total production needed to fulfill all orders and demand. As such, we summed the order and demand quantities and divided by the number of factories to get the capacity needed at each factory. This assumes each factory will share production equally. Our model timeframe spans 10 days; however, it is not a rolling model, so it is only concerned with the orders and demand that come in on the *first day* of the model. To account for this in the production, we can divide the capacity of each factory by the number of days in the model. This evenly distributes capacity across the entire model timeframe, which is not quite what we are after. By doing this, the factories will not have enough capacity in the first few days of the model to meet the demand and orders, given that the associated service level lead times range from three days to seven days.

We need to distribute most of the capacity to the first few days of the model so that demand can be satisfied. In essence, the model timeframe is set at 10 days to account for the delivery of the products; the purpose is not to spread production across the entire model evenly. Consequently, we multiply our daily factory capacity by a *production factor* to increase the value. This can be set anywhere from 1.5 to 3, depending on the level of capacity shortage or surplus desired. Finally, the resulting production capacity values are randomized by +/- 150 capacity units.

We next set the kitting part stock value. Recall that this is the daily supply of parts used in production of the Kitting SKUs. Each part can have a different inventory level. To simplify matters slightly, we kept the daily stock levels uniform across the model timeframe. Our goal was to set the levels of the part stocks to match the needed capacity to fulfill all demand/orders. For each part, we first checked the Kitting SKU BOM's to determine the quantity used in each product. We used this to estimate the total quantity of each part needed to fulfill the requested demand/orders for Kitting SKUs. For example, given that we have three Kitting SKUs, if Part1 is used in both Kitting SKU1 and Kitting SKU3 and the total demand/orders of Kitting SKUs is 12,000 units, then to get the quantity of Part1, we multiply 12,000 by $\frac{2}{3}$, resulting in 8,000 units of Part1 needed. Now, similar to the production capacity, this is divided by the number of factories and the number of days in the model timeframe. To account for production occurring in the earlier time stages of the model, we multiply this by a *kSKU factor* to get the

final value of each part. To finish, we vary this by +/- 50 units to randomize the resulting values of parts.

Finally, we need to set the values for the Merging SKU stock. This is done in a similar fashion as setting the part stock values. We first calculate the total Merging SKU quantity for orders/demand. We then divide this by the number of merging centers and the number of days in the model. Finally, we multiply this value by an *mSKU factor* to adjust upwards the daily value. Now, since the merging will occur after production, this factor should be set slightly lower than the *Production* and *kSKU factors*. This value is then varied by +/- 100 units to randomize.

The table below summarizes the formulation of each of these parameters.

Parameter	Value	Random Range
Order Quantity (product/ser. level)	Average = 650	+/- 100
Demand Quantity (product/ser. level)	Average = 650	+/- 150
Production Capacity (per day, factory)	$\frac{\sum \text{Requested kSKUs}}{\text{Days} * \# \text{ Factories} * \text{Production Factor}}$	+/- 150
Stock of Parts (per day, part, factory)	$\frac{\sum \text{Parts Needed (based on kSKUs requested, BOMs)}}{\text{Days} * \# \text{ Factories} * \text{kSKU Factor}}$	+/- 50
Stock of mSKUs (per day, mSKU, merging center)	$\frac{\sum \text{Requested mSKUs}}{\text{Days} * \# \text{ M. Centers} * \# \text{ mSKUs} * \text{mSKU Factor}}$	+/- 100

Table 3.7: Formulation of Parameter Values

3.4. Model Specifications

This section provides the specifications of the problem solution. The model is solved efficiently within Xpress, especially considering the large size of the dataset in the model. The average solution time is usually less than a minute, although some scenarios require longer solving time.

We first analyze the model specifications for the base setup of the model, using the data size discussed in the previous section. This base model has over 8,000 decision variables; 2,700 of which are restricted as binary variables. As part of the solution process, the model is first preprocessed (or presolved) to tighten constraints and remove any redundancies. The following table specifies the size of the problem matrix in both instances. The preprocessing stage cut the number of rows (constraints) and columns (decision variables) by almost half.

	Initial Value	Value after Presolve
Rows	6,481	3,814
Columns	8,272	4,145
Non-zero Elements	29,713	15,184

Table 3.8: Model Size – Xpress Solver

After the preprocessing phase, the mixed integer constraints are removed, and the subsequent Linear Program is solved using the Simplex (dual) method. This took 1,956 iterations. The optimal solution to the LP relaxation is found in less than one computational second. 3

In the next step of the solving process, the optimizer algorithm adds cuts to the problem (valid inequalities that cut off fractional solutions), thus drawing the

LP relaxation closer to the convex hull of integer solutions and improving the bound provided by the relaxation. For our problem, 190 cuts were made (190 rows added to problem matrix). Finally, a solution is found by using a branch-and-bound algorithm. The following graph shows the node search for the optimal solution. The integer solution is found at Node 73 in 5 seconds of computational time.

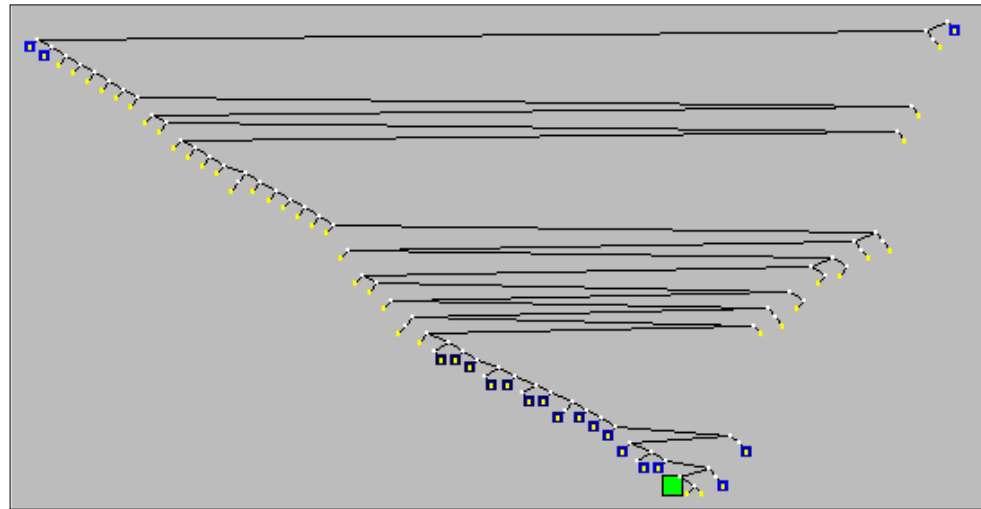


Figure 3.7: Branch and Bound Search to Identify MIP Solution

As we discovered during the experimental analysis, certain business scenarios of the model take considerably longer to solve in Xpress. For instance, in the case where capacity resources are very scarce and the commitment of demand is binary, the optimizer needs several minutes to find an optimal integer solution. While the model size is the same as the base run, the constraints are much tighter, so the solution is harder to find. The LP relaxation problem was solved in 2,506 iterations of the dual Simplex method. The optimizer added 174 cuts to the problem. Finally, the problem was solved using branch-and-bound search

enumeration of 12,025 nodes. The search found 8 feasible solutions before the optimal value was identified. The following table gives the differences in the base run and this run of increased computational complexity.

	Base Run	Complex Run
LP Relaxation	1,956 iterations	2,506 iterations
Cuts	190 cuts	174 cuts
Branch-and-Bound	73 nodes 1 integer solution 5 computational seconds	12,025 nodes 9 integer solutions found 200 computational seconds

Table 3.9: Comparison of Computational Complexity of Two Runs

Xpress enables customization of the cutting plane strategy, branch-and-bound algorithm and other parameters in the solution process. However, for even the most complex setup for our model, the problem was solved within minutes, so we did not experiment with these settings to search for computational improvements.

4. Study of Experiments

In this chapter, we first discuss the verification process of the model. Next, we go through the sensitivity analysis that we performed on the model. Finally, we discuss the various experiments conducted – their purpose, their required updates to the model formulation/data, and their results.

4.1. Verification of Model

Before experiments were conducted, we verified the model to check that it was setup properly. Verification is the process of ensuring that the model and all its components meet the requirements and specifications of the design. The goal of the verification testing phase is to find all errors and fix the underlying causes.

To verify the model, we conducted a series of tests to make sure it acted in the manner we intended. We developed three scenarios for this testing. Through the course of our verification testing, minor issues with the model formulation were discovered and fixed. The tests were re-run to ensure the issues were resolved.

Test 1: Verify flow of parts, Kitting SKUs, Merging SKUs

For this test, we checked to make sure that the flow of parts and products was correct throughout the model timeframe. An important aspect of this test was guaranteeing that the schedule of departures, transportation times and arrivals were calculated based on the correct lead time parameters. Additionally, material conservation at each step of the supply chain was analyzed for accuracy. This

meant checking that the production and inventory levels were correct, given the quantity shipped for orders, and so on.

For this scenario, we set an unlimited production capacity at the factories, and high levels of the supply of parts and Merging SKUs. The initial inventories of finished Kitting SKUs were kept low to force production at the factories.

We analyzed the solution values to verify the results were as expected. We calculated the expected inventory levels and production based on the order and demand shipments and verified material was moving appropriately at each factory and merging center. In addition, we checked each order/demand arrival to ensure the quantity and date were correct. We uncovered a slight issue with demand orders, in that they were all arriving one day *after* the expected date for the corresponding service level. This was attributed to an error in the constraint for service level definition; this was fixed and re-tested to satisfaction.

Test 2: Delayed Deliveries of Orders

In this test, we wanted to ensure that the model was correctly handling late deliveries for orders. It should allow orders to be delivered late and should assess the correct due date violation penalty. Additionally, we wanted to make sure the model was making the expected trade-offs between demand commitment and order scheduling.

For this scenario, we limited the production levels at the factories, and reduced the supply of kitting parts and Merging SKUs. This would force some orders to be delivered late, and some demand to go uncommitted.

After running the model, we checked the model to verify the scenario results. The model correctly committed higher profit demand requests in exchange for delivering orders late. However, one issue was uncovered in this test. Orders that went undelivered for the entire model timeframe were not being assessed a due date penalty. We added a term in the constraint for due date violation definition to account for undelivered orders to resolve this issue.

Test 3: Calculation of Profits and Costs

In our final test, we wanted to verify that model was correctly calculating the profits and costs of each order and demand. We needed to ensure that the correct profit margins were used for each product and service level. Furthermore, the transportation costs should reflect the right fixed and variable costs for each shipping mode. And again, we wanted to verify the due date violations were correct for all products and service levels.

We ran this trial and verified the results with our expectations. No errors were found with the model formulation; each order had the proper values for profits, transportation costs and due date violation penalties.

4.2. Sensitivity Analysis

After the model had been verified using the generated data, we carried out various sensitivity analyses to test the capabilities of the model. By running the model under different conditions to simulate various business scenarios, we observe the behavior and power of the model.

The first step was to create a base scenario of the model. This is used as a benchmark to check performance and decisions of later iterations. In subsequent runs, the majority of the data parameters have the same values as the base run, so that the impact of varying one or two parameters can be identified and analyzed. We tested scenarios such as variances in profit margins, reduced capacity, and part shortages. The details of each test and the ensuing results are presented next.

4.2.1. Base Setup of Model

The base run of the model is intended to serve as a point of reference for future trials. This run is based on the expected, or standard, business operating environment for a MTO/CTO company. In this scenario, the resource availability, including both production capacity and parts, roughly matches the incoming orders and forecast demand. As discussed earlier, these values are generated randomly within a set range, so in some trials there might be a slight shortage of resources, while other trials might result in a slight excess of resources. This is done to mimic the uncertainty of production planning and demand forecasting.

Next, we set up the formulation of critical data for the base trial, given the business scenario. The following table provides the approximate range of values for the key parameters.

Parameter	Approximate Range of Values
Order Quantity (per order of product/service level)	550 – 750
Demand Quantity (per demand of product/service level)	500 – 800
Production Capacity (per day, per factory)	1,000 – 1,350
Stock of mSKUs (per day, per mSKU, per merging center)	85 – 250
Stock of Parts (per day, per part, per factory)	350 – 1200
Profit Margins	Gold Service Level: 5% Silver Service Level: 2% Bronze Service Level: 0%
Objective Value Weights	Order revenue: 0.5 Order delivery costs: 0.5 Demand revenue: 0.5 Demand delivery costs: 0.5 Due date violation penalty: 1.0

Table 4.1: Range of Values for Parameters in Base Model

Thirty trials were run to get a complete and accurate set of data to use for analysis of the base model. Key results are detailed in the following table.

Output	Average Value
Objective Function	1,225,632
Profits	\$ 4,291,330
Orders	\$ 2,444,865
Demand	\$ 1,846,465
Costs	\$ 1,787,663
Order Delivery	\$ 996,839
Due Date Violations	\$ 52,402
Demand Delivery	\$ 738,423
Demand Commitment	73.0%
Gold Service Level	48.7%
Silver Service Level	80.5%
Bronze Service Level	89.6%
kSKU1	87.0%
kSKU2	87.3%
kSKU3	89.6%
mSKU1	49.2%
mSKU2	58.4%

Table 4.2: Results of Base Trial

We can see that a fair percentage of overall demand was accepted (73%). Within demand, a much higher percentage of lower service levels (Silver and Bronze) were committed, at 80.5% and 89.6%, respectively. This makes sense in this scenario, as there is greater flexibility in production resources and shipping modes for orders with longer lead times. For a shortened lead time, as with the Gold service level, fewer resources are available to share with demand, resulting in a lower commitment percentage (48.7%). Although this figure might seem low for some companies, remember that it is based entirely on the level of resources provided as input for this base model. Naturally, a company with higher availability of production resources would have higher commitment levels.

As for orders, we can see that a limited number of orders are delivered late, resulting in a due date violation. However, this number is relatively small, which makes sense considering that the resources were fairly balanced with the orders and demand.

4.2.2. Sensitivity Analysis of Profit Margins

In this test, we wanted to see how changing the profit margins would affect the overall commitment decisions. Therefore, we kept the demand and order sizing the same, as well as the production capacity and stock of parts and Merging SKUs. The only parameter that was changed was the profit margin. This table details the changes. The profit margin includes the service level and product profit margin.

Parameter	Base Value	New Value
Gold Profit Margin	30%	45%
Silver Profit Margin	27%	30%
Bronze Profit Margin	25%	25%

Table 4.3: Parameter Settings for Sensitivity Analysis of Profit Margins

This scenario is useful to show the case in which a company can charge an extremely high markup for a first-class service level. In some markets, the customers can be segmented as such and products with faster delivery can achieve a considerable premium over basic delivery.

We expect that the model will change the allocation of resources in the base model to account for the additional profits that could be made from Gold service level products. Consequently, the demand commitment for Gold products should increase, while the commitment for Bronze products might decrease. Additionally, more orders might be delivered late to compensate for the shift in resource allocation.

As with before, 30 trials were run in this test. The results are detailed in the following table.

Output	Base Scenario: Average Values	Inc. Profit Marg.: Average Values	Difference
Objective Function	1,225,632	1,607,658	31.2%
Profits	\$ 4,291,330	\$ 5,185,370	20.8%
Orders	\$ 2,444,865	\$ 2,906,839	18.9%
Demand	\$ 1,846,465	\$ 2,278,531	23.4%
Costs	\$ 1,787,663	\$ 1,890,781	5.8%
Order Delivery	\$ 996,839	\$ 1,013,853	1.7%
Due Date Violations	\$ 52,402	\$ 79,656	51.3%
Demand Delivery	\$ 783,423	\$ 797,656	8.0%
Demand Commitment	73.0%	76.9%	5.4%
Gold Service Level	48.7%	68.6%	40.9%
Silver Service Level	80.5%	78.5%	-2.5%
Bronze Service Level	89.6%	83.8%	-6.5%
kSKU1	87.0%	87.2%	0.2%
kSKU2	87.3%	87.3%	0.0%
kSKU3	83.0%	87.7%	5.7%
mSKU1	49.2%	56.1%	14.1%
mSKU2	58.4%	66.5%	13.8%

Table 4.4: Results of Sensitivity Analysis of Profit Margins

From Table 4.4, we can observe that as predicted, the commitment levels for the Gold service level products have increased by 40.9%, from 48.7% to 68.6%. The model re-allocates the capacity to include more production for the Gold products, increasing the overall profits. In doing so, the commitment levels for the Silver and Bronze levels have dropped somewhat. Additionally, the due date penalty jumped over 50% to account for the shift in allocation. Clearly, the model will assign a late delivery to certain low profit orders if that means it then can accept other higher profit demand. Additionally, the delivery costs for orders and forecast demand increased. This indicates that the model delayed production and used alternate, faster delivery modes to ship out some orders.

While the profits also increased, this is due more to the fact that the company is earning much more for products with the Gold service level. Although, we can see that the increase in profits for demand (23.4%) is slightly higher than the increase for orders (18.9%). This can be explained due to the higher commitment level, overall, of the demand in this new model scenario.

When we analyze the demand commitment classified by product, we can see insignificant changes for the Kitting SKU demand, with slightly higher increases for the Merging SKU demand. Since the commitment levels for demand of Kitting SKUs were already relatively high (mid 80% range), this would indicate that the commitment simply shifted towards the Gold, from Silver and Bronze within each Kitting SKU demand. Thus, the overall commitment for each demanded Kitting SKU would remain the same.

As for the Merging SKUs, only about 50% of the demand was committed in the base model, which is comparatively lower. As such, with the new increased profit margins, a higher level of total demand for each Merging SKU was reserved. This would indicate that the capacity from orders was shifted to account for the increase in demand acceptance.

4.2.3. Sensitivity Analysis of Production Capacity

We next wanted to test the sensitivity of the model relating to capacity. This would correspond to the scenario in which demand for a company's products far outweighs the resources to meet demand. For testing this, we kept the order and demand quantities the same, but severely reduced the production capacity and the

inventory-on-hand for parts and Merging SKUs, as seen in the following table. The stock of Merging SKUs is not reduced as much as the production capacity and the stock of parts. This is because the base model had fairly tight capacity for the Merging SKUs, and any greater of a reduction would cause an imbalance in the capacity shortage analysis.

Parameter	Reduced Capacity Scenario
Order Quantity	Same as base model
Demand Quantity	Same as base model
Production Capacity (per day, per factory)	50% of base model value
Stock of mSKUs (per day, per mSKU, per merging center)	66% of base model value
Stock of Parts (per day, per part, per factory)	50% of base model value
Profit Margins	Same as base model

Table 4.5: Parameter Settings for Sensitivity Analysis of Production Capacity

The general expectation is that the model will change commitment and resource allocation to favor those products with the highest profit margins. We also expect due date violations to increase, and the overall commitment levels for demand to decrease. This scenario was run over 10 trials. The comparison of the results with the base scenario is detailed in the following table.

Output	Base Scenario: Average Values	Red. Capacity: Average Values	Difference
Objective Function	1,225,632	787,163	-35.8%
Profits	\$ 4,291,330	\$ 3,253,940	-24.2%
Orders	\$ 2,444,865	\$ 2,379,792	-2.7%
Demand	\$ 1,846,465	\$ 874,147	-52.7%
Costs	\$ 1,787,663	\$ 1,498,809	-16.2%
Order Delivery	\$ 996,839	\$ 979,418	-1.7%
Due Date Violations	\$ 52,402	\$ 180,804	245.0%
Demand Delivery	\$ 738,423	\$ 338,587	-54.1%
Demand Commitment	73.0%	34.1%	-53.3%
Gold Service Level	48.7%	12.2%	-74.9%
Silver Service Level	80.5%	27.8%	-65.5%
Bronze Service Level	89.6%	62.1%	-30.7%
kSKU1	87.0%	21.9%	-74.8%
kSKU2	87.3%	55.2%	-36.7%
kSKU3	83.0%	11.9%	-85.7%
mSKU1	49.2%	34.1%	-30.6%
mSKU2	58.4%	46.0%	-21.3%

Table 4.6: Results of Sensitivity Analysis of Production Capacity

The effects of reduced resources are clear. The model was able only to commit 34.1% of demand, a drop of over 50% from the base scenario. We can see large decreases in profits as well. Tellingly, the due date violations increased by 245%, which clearly indicates the shortage of resources all around.

In comparing the demand commitment for the various service levels, we see that Gold decreased 74.9%, Silver 65.5%, and Bronze 30.7%. This result is somewhat curious, because Gold products have the highest profit margins, yet suffer the greatest decrease in commitment. However, we can surmise that given the extremely tight lead time with the Gold service level, the production resources and part inventory were so reduced they simply could not produce both the Gold service level orders and the Gold service level demand. More flexibility later in the

model time frame enabled the model to adjust and re-allocate resources so that more of the demand could be accepted for the lower service levels.

The interesting aspect of this sensitivity analysis is to see where the model shifted capacity and delivery schedules. The commitment percentages for the various products all decreased, but not in comparable fashion. If we look at the profit margins for each of the products, the ones with the higher profit margins decrease the least. So, the model is making decisions to re-allocate capacity from lower profit margin products to those in which it can achieve a higher profit. The table below shows the results based on profit margins of products.

Product	Difference from Base Scenario	Profit Margin (in relation to mSKU1)
Kitting SKU1	-74.8%	127%
Kitting SKU2	-36.7%	184%
Kitting SKU3	-85.7%	105%
Merging SKU1	-30.6%	100%
Merging SKU2	-21.6%	111%

Table 4.7: Profit Margin Results of Sensitivity Analysis of Production Capacity

The profit margins are evaluated against the lowest profit margin product (Merging SKU1). We can see that the profit margin for Kitting SKU2 is almost twice that of Merging SKU1. This relationship holds across all service levels for the products (e.g.: Bronze service level profit margin for Kitting SKU1 is 127% of the Bronze service level profit margin for Merging SKU 1, and so forth). Based on this data, it is clear to see how the model re-allocated resources. When we compare the Kitting SKUs (which share common production capacity and parts), we can see that the model is allocating resources first for demand of higher profit margins.

Kitting SKU2, with the highest profit margin only dropped 36%, while the other two products dropped 75% and 85%, respectively. Available production capacity and parts go toward the manufacturing of Kitting SKU2 first. This is a considerable result – it shows that the model is allocating resources for the higher profit products under limited capacity situations. It should be noted that the profit margins of Kitting SKU1 and Kitting SKU3 for the Gold Service Level are higher than the profit margin of Kitting SKU2 for the Bronze Service Level. This helps explain why there is not a total shift away from these lower profit products.

4.2.4. Sensitivity Analysis of Part Shortage (Unique Part)

In this trial, we wanted to test the effect of a shortage of a part on overall commitment and resource scheduling. We chose Part4, which is used solely in Kitting SKU2, and reduced the stock supply value by 50% from the base scenario. All other parameters, including order and demand quantities, production capacity and the stock of other parts, were kept at the same values as the base scenario.

Because this part is used only in one product, we expect that particular product to have reduced commitment levels and increased due date violations for orders. This scenario was run over 30 trials, and the data was collected below:

Output	Base Scenario: Average Values	Decreased Stock of a Unique Part	Difference
Objective Function	1,225,632	1,125,758	-8.1%
Profits	\$ 4,291,330	\$ 4,071,801	-5.1%
Orders	\$ 2,444,865	\$ 2,444,865	0.0%
Demand	\$ 1,846,465	\$ 1,626,936	-11.9%
Costs	\$ 1,787,663	\$ 1,717,131	-3.9%
Order Delivery	\$ 996,839	\$ 966,237	-3.1%
Due Date Violations	\$ 52,402	\$ 103,154	96.9%
Demand Delivery	\$ 738,423	\$ 647,740	-12.3%
Demand Commitment	73.0%	67.4%	-7.7%
Gold Service Level	48.7%	42.0%	-13.7%
Silver Service Level	80.5%	73.0%	-9.3%
Bronze Service Level	89.6%	87.0%	-2.9%
kSKU1	87.0%	87.2%	0.2%
kSKU2	87.3%	57.5%	-34.1%
kSKU3	83.0%	85.4%	2.8%
mSKU1	49.2%	49.2%	0.0%
mSKU2	58.4%	58.4%	0.0%

Table 4.8: Results of Sensitivity Analysis of Part Shortage (Unique Part)

We can see that as predicted, the commitment of future demand for Kitting SKU2 is greatly reduced in this scenario (by 34%). Additionally, the due date violations have almost doubled, which reflects that more orders involving Kitting SKU2 cannot be shipped in time with the reduced resources.

When analyzing the service level commitments, there is a definite drop for the Gold service level (13.7%), and less profound drops for the other two service levels. As demand cannot be delivered late, we can surmise that much of the demand involving Kitting SKU2 was not committed at all. The drops are lower for the lower service levels, which indicates that with a longer planning horizon, the model can effectively distribute resources so that more demand can be met (through delaying production of orders of similar configurations and delivering orders late).

It is interesting to note the commitment for Kitting SKU1 and SKU3 increased in the run (by 0.2% and 2.8%, respectively). The shortage of the part limited the production of Kitting SKU2. Therefore, the common resources (production capacity and other part availability) can be reallocated for production of Kitting SKU1 and SKU3.

4.2.5. Sensitivity Analysis of Part Shortage (Common Part)

This analysis is similar to the prior case. However, in this version, the stock of the part that is reduced is one that is commonly shared across all Kitting SKUs. We kept the demand and order quantities the same as in the base scenario, as well as the production capacity and the stock quantities of other parts. For Part3, we reduced the level of stock by 40%. This part is on the BOM for all three Kitting SKUs.

We expect that overall due date violations would increase and the demand commitment would be reduced somewhat. The key aspect of this test is to analyze specifically how the profit margins of these products affect the commitment levels. As the part is used in all Kitting SKUs, we expect that the lower profit margin products will be reduced at greater levels than those products with higher profit margins. We ran this scenario over 30 trials, and recorded the results in the below table.

Output	Base Scenario: Average Values	Decreased Stock of a Common Part	Difference
Objective Function	1,225,632	1,048,062	-14.5%
Profits	\$ 4,291,330	\$ 3,863,834	-10.0%
Orders	\$ 2,444,865	\$ 2,444,865	0.0%
Demand	\$ 1,846,465	\$ 1,418,970	-23.2%
Costs	\$ 1,787,663	\$ 1,004,692	-7.3%
Order Delivery	\$ 996,839	\$ 1,004,692	-0.8%
Due Date Violations	\$ 52,402	\$ 110,637	111.1%
Demand Delivery	\$ 738,423	\$ 541,744	-26.6%
Demand Commitment	73.0%	54.3%	-25.6%
Gold Service Level	48.7%	28.0%	-42.4%
Silver Service Level	80.5%	53.1%	-34.0%
Bronze Service Level	89.6%	81.7%	-8.8%
kSKU1	87.0%	56.0%	-35.6%
kSKU2	87.3%	87.3%	-0.0%
kSKU3	83.0%	19.9%	-76.0%
mSKU1	49.2%	49.2%	0.0%
mSKU2	58.4%	58.4%	0.0%

Table 4.9: Results of Sensitivity Analysis of Part Shortage (Common Part)

We can see that the overall demand commitment is reduced (by 25%) and the due date violations are substantially higher (111%) due to the part shortage of this common part. The model cannot commit resources for all the forecast demand, so it must determine which demand should be committed. Based on these results, it took capacity from some orders, as seen in the late deliveries. More significant is that of the demand that was committed, the majority is for orders involving Kitting SKU2. This product saw no effect of the reduced part stock, while Kitting SKU1 and Kitting SKU3 were significantly reduced in commitment percentages (35% and 76%, respectively). These results make sense when we remember that Kitting SKU2 has a much higher profit margin than the other two products. The following

table highlights the differences. In this version, the value of profit margins is in terms of Kitting SKU2.

Product	Difference from Base Scenario	Profit Margin (in relation to kSKU 2)
Kitting SKU1	-35.6%	69%
Kitting SKU2	-0.0%	100%
Kitting SKU3	-76.0%	57%

Table 4.10: Profit Margin Results of Sensitivity Analysis of Part Shortage (Common Part)

The model was still able to meet the same level of demand for Kitting SKU2. However, with the part shortage, the same levels could not be committed for the other two products. Because Kitting SKU2 had the highest profit margin, resources were reserved for it first. Next, it fulfilled demand for Kitting SKU1, then finally Kitting SKU3. Some capacity is still reserved for these products, because the profit margins at the Gold level are slightly higher than those of Kitting SKU2 at a Bronze service level, and so on.

4.3. Experiment 1: Commitment Policy

In this experiment, we test the demand commitment policy. In the base model, demand can be committed as a percentage, so 45% of a particular demand could be reserved. Because the demand values are aggregate, this partial commitment corresponds to the scenario where the company could reserve resources to produce most of the requested demand of that product, but not the full expected demand.

We now want to test the policy where the demand is either committed fully or not at all for a particular product configuration and service level. In some cases,

this would be a more accurate measure of the demand resource planning for a company. For instance, in the scenario in which a company conducts sales on-line, each business day, decisions must be made regarding which service levels should be offered for the various products. If a company cannot commit to producing a certain product within the time frame of a premium service level, they would not even offer that option to the customers. Our base model would allow some demand to be committed, which is unacceptable in this business scenario. We are assuming here, of course, that the on-line sales system is incapable or not setup for real-time updates of product/service line availability throughout the day.

The model can be modified easily to account for this new constraint. Instead of modeling the commitment of demand as a continuous variable between 0 and 1, we now further constrain it as a binary variable. No other constraints in the model need to be altered.

Upon analysis of the base model setup, it is noted that the majority of demand is accepted at values either very close to 100% or very close to 0%. Thus, the shift to 0% or 100% will not yield significant changes. This new policy of all-or-nothing commitment would have the greatest impact in a business scenario in which the majority of demand was committed at percentages of 25% to 75%. So, in our test of the new policy, we reduced the capacity of part stock, production, and Merging SKUs each by 20% (by altering the *kSKU Factor*, *Production Factor*, and *mSKU Factor* in the data formulas) so that demand was reserved at varying levels (more so at least than the base setup).

Once the model has been updated, we run it with the same dataset size and values as the base model to gauge the results of this new constraint. We expect that overall commitment of demand would be reduced, or possibly more orders would be delivered late to accommodate increased demand quantity commitment. We ran the new model over 10 trials, and compared the results with the base setup of the model.

Output	Base Scenario: Average Values	Exp. 1: Commit 0 or 100%	Difference
Objective Function	1,037,306	1,034,643	-0.3%
Profits	\$ 3,946,268	\$ 3,961,333	0.4%
Orders	\$ 2,412,416	\$ 2,412,416	0.0%
Demand	\$ 1,533,852	\$ 1,548,917	1.0%
Costs	\$ 1,728,828	\$ 1,744,388	0.9%
Order Delivery	\$ 978,375	\$ 977,906	0.0%
Due Date Violations	\$ 142,827	\$ 147,659	3.4%
Demand Delivery	\$ 607,626	\$ 618,824	1.8%
Demand Commitment	59.1%	59.8%	1.2%
Gold Service Level	29.2%	29.0%	-0.7%
Silver Service Level	63.8%	66.0%	3.4%
Bronze Service Level	84.0%	84.2%	0.2%
kSKU1	71.8%	73.4%	2.2%
kSKU2	87.3%	86.8%	-0.6%
kSKU3	56.3%	58.1%	3.2%
mSKU1	30.3%	29.8%	-1.4%
mSKU2	50.5%	51.7%	2.3%

Table 4.11: Results of Experiment 1: Commitment Policy

Though the new policy does not yield highly significant changes, there are several interesting results. The objective function of the model decreased by 0.3%. While the profits actually increased slightly with the new policy, the high cost of the due date violations outweighed any additional earnings. The increase of profits looks to be related to the increase in reserved demand (1.2%). It is possible that

much of the demand was pushed up to full commitment from a high fractional value of commitment. However, we can see that this had a profound effect on the orders. The due date violations increased by 3.4% to compensate for the additional resources consumed by demand.

Since the same data was used in the trials, we can analyze the results to see specifically where the model made shifts with this new policy. The following are the significant changes in the demand settings:

Demand Details	% Committed	Effect on Orders
ID: D16 Gold Service Level 222 of kSKU2	Base: 4.5% 0-1: 0%	Comparable order delivery delayed slightly; added due date violation.
ID: D35 Silver Service Level 279 of mSKU1	Base: 40.5% 0-1: 0%	The new policy model (0-1 demand commitment) had reduced due date violations (27%) for Silver/Bronze Service Level of mSKU1 as more resources were available for orders since demand not reserved.
ID: D36 Bronze Service Level 294 of mSKU1	Base: 84.0% 0-1: 100%	See above. Although additional demand was committed for mSKU1 for the Bronze Service Level, much more went uncommitted for the Silver Service Level, creating an overall effect of reduced due date violations for similar orders.
ID: D43 Gold Service Level 255 of mSKU2	Base: 71.8% 0-1: 100%	The new policy model (0-1 demand) had increased due date violations (116%) as well as increased delivery costs (7.6%) for Gold/Silver/Bronze orders of mSKU2 when the model committed 100% of this demand, rather than 71.8%.

Table 4.12: Analysis of Effect on Orders for Commitment Policy Experiment

In conclusion, if a manufacturing facility has plentiful capacity and part inventory, the policy of reserving demand at 0 or 100% is insignificant. However, in the scenarios in which capacity is limited, forcing the model to reserve all or

none of the demand will result in higher due date violation penalties and delivery costs for accepted orders.

4.4. Experiment 2: Service Level Policy Analysis

In this experiment, we test the very notion of using service levels for order delivery. As we discussed earlier, the majority of companies use a policy in which the customer selects the desired shipment date and mode. In our model, we have been testing the effects of allowing a customer to pick a service level, which corresponds to the arrival date of the product. The manufacturer could then select the shipment mode and schedule that best fit its production needs.

We want to see the effects of this new service level policy with the standard delivery setting policy. To do this, we created a model that used the standard policy whereby the delivery mode and schedule is set by the customer.

Our first step was to alter our model so that each service level now corresponds to a specific delivery mode for both demand and orders. Therefore, the model does not make any decisions on how or when to ship each order. The only flexibility in delivery is through delaying shipment and accepting a due date violation penalty. The following changes were made to the model formulation:

Parameters

Indices - No Change

Orders

Added :

$om^{l,k}$ = Transportation mode l for order k

Forecast Demand

Added :

$dm^{l,d}$ = Transportation mode l for demand d

Costs – No Change

Production/Merging/Delivery - No Change

Inventory - No Change

Decision Variables

Orders/Demand - No Change

Costs/Profits - No Change

Production/Inventory - No Change

Objective Function :

Maximize Profit - No Change

Subject to:

(1) *Profits and Costs Definition - No Change*

(2) *Order Delivery*

Added :

$$(2.7) \quad LO^{k,l,t} \leq om^{l,k} \text{ for all } k \in K, l \in L, t \in T$$

Each order must be delivered by its associated delivery mode

(3) *Demand Delivery*

Added :

$$(3.6) \quad LD^{d,l,t} \leq dm^{l,k} \text{ for all } d \in D, l \in L, t \in T$$

Each demand must be delivered by its associated delivery mode

(4) *Material Conservation - No Changes*

We then set up the data for the model. The Gold service level was associated with the fastest delivery mode (1-Day Air), the Silver with the next

fastest (2-Day Air) and the Bronze with the slowest delivery mode (3-Day Ground).

All the other data values were generated exactly as with the base model.

Next, we ran the model under the same scenario as the base model for 30 trials. We expect that since the model cannot determine the delivery schedule, perhaps more orders will be delivered late or some demand will go uncommitted.

The following table outlines the results.

Output	Base Scenario: Average Values	Exp. 2: Service Level = Del. Mode	Difference
Objective Function	1,225,632	1,156,759	-5.6%
Profits	\$ 4,291,330	\$ 4,282,665	-0.2%
Orders	\$ 2,444,865	\$ 2,464,047	0.8%
Demand	\$ 1,846,465	\$ 1,818,618	-1.5%
Costs	\$ 1,787,663	\$ 1,695,273	-5.2%
Order Delivery	\$ 996,839	\$ 838,053	-15.9%
Due Date Violations	\$ 52,402	\$ 273,874	422.6%
Demand Delivery	\$ 738,423	\$ 583,346	-21.0%
Profits – Del. Costs	\$2,503,667	\$2,587,392	3.3%
Demand Commitment	73.0%	72.3%	-0.9%
Gold Service Level	48.7%	49.1%	0.8%
Silver Service Level	80.5%	81.7%	1.5%
Bronze Service Level	89.6%	85.9%	-4.1%
kSKU1	87.0%	87.0%	0.0%
kSKU2	87.3%	87.3%	0.0%
kSKU3	83.0%	82.2%	-1.0%
mSKU1	49.2%	49.0%	-0.4%
mSKU2	58.4%	55.5%	-5.0%

Table 4.13: Results of Experiment 2: Service Level Policy

So, as expected, the profits from demand drop slightly, but of greater significance is the decrease of costs related to delivery. The cost to deliver orders and demand has decreased (15.9% and 21.0%, respectively), which is intuitive. The model cannot delay production and then expedite delivery through a faster mode, which would increase the delivery costs, as is the case in the base model.

Thus, the delivery costs are reduced in this new model. Also of note is the sharp increase in due date violations (422.6%). Clearly, since the model has less flexibility in production scheduling and delivery, more orders are delivered late when higher profit margin demand is committed.

The net profits (excluding due date violations) actually increased in the model when the standard service level policy was used. Upon closer inspection, this can be attributed to the high costs of the due date violation. In the base run of the model, the objective function weights are equal for the order costs and profits and the demand costs and profits. The penalty for due date violations is double the weight of these other four factors. We can see how this results in the model seeking specifically to set the production to minimize these penalties wherever possible. We ran these trials again, but this time set the due date violation equal to the other weights of the objective function.

Output	Base Scenario: Average Values	Exp. 2: Service Level = Del. Mode	Difference
Objective Function	1,345,615	1,156,759	-2.9%
Profits	\$ 4,457,979	\$ 4,473,106	0.3%
Orders	\$ 2,449,015	\$ 2,449,015	0.0%
Demand	\$ 2,008,964	\$ 2,024,090	0.8%
Costs	\$ 1,628,816	\$ 1,693,720	4.0%
Order Delivery	\$ 792,975	\$ 834,571	5.2%
Due Date Violations	\$ 315,275	\$ 378,600	20.1%
Demand Delivery	\$ 678,203	\$ 669,849	-1.2%
Profits – Del. Costs	\$ 2,829,163	\$2,779,386	-1.8%
Demand Commitment	80.3%	80.8%	0.7%
Gold Service Level	62.8%	65.2%	3.8%
Silver Service Level	87.6%	89.3%	1.9%
Bronze Service Level	90.3%	87.8%	-2.7%
kSKU 1	87.2%	87.2%	0.0%
kSKU 2	87.3%	87.2%	-0.1%
kSKU 3	89.2%	89.2%	0.0%
mSKU 1	65.6%	67.8%	3.3%
mSKU 2	72.2%	72.9%	1.1%

Table 4.14: Results of Experiment 2: Service Level Policy (Equal Objective Weights)

In this run, the difference in regards to due date violation is not as pronounced. The penalty increased 20.1%, as compared to 422% in the earlier run. We can see that when due date violations are not weighted as heavily, the model will deliver more orders late in exchange for accepting additional demand or choosing a less expensive delivery mode for the base run. In this case, the delivery costs actually increased using the model with the standard delivery policy. We surmise that the model was able to shift delivery from some of the faster (and more expensive) modes to slower modes, in cases where production could be completed earlier.

4.5. Experiment 3: Re-pointing of Merging Center

In this experiment, we test the policy relating to merging center re-assignment. In the base model, incoming orders are assigned to the closest merging center based on their geographic location. We want to measure the effects when the model actually determines the merging center to serve the customer.

We expect that the new model will align orders to the closest merging centers in the majority of cases. However, for the scenario that production capacity is severely limited, it might make sense to produce and merge the order at another merging center and then ship from there to the customer. In this case, the shipping time would be longer, but it might outweigh delays in production. This concept of re-assigning order/demand scheduling is defined as *re-pointing*.

Implementing this new policy requires modification of several parameters, decision variables and constraints in the model. A new parameter was created for the order/demand location (e.g.: East, Midwest, South, etc.). As input to the model, the travel distance from each of these locations to the various merging centers must be specified. This is then added to the delivery time for a specific mode to get the overall lead time for shipment from a merging center to the customer location. In cases where the merging center assigned is the closest center, no additional time is added.

The following is the updated formulation of the model; the numbering scheme of the constraints has not changed for this experiment, but updates were made in the formulation of certain constraints, as marked below.

Parameters

Indices

Added :

c = Order location

Orders

Changed :

$ol^{c,k}$ = Location c of order k

Forecast Demand

Changed :

$dl^{c,d}$ = Location c of demand d

Costs – No Change

Production/Merging/Delivery

Added :

$ll^{c,m}$ = Lead time to transfer from merging center m to customer location c

Inventory - No Change

Decision Variables

Orders/Demand

Changed (added parameter m):

$LO^{k,l,m,t}$ = Delivery status for order k by transportation mode l from merging center m in time period t ; *binary*

$LD^{d,l,m,t}$ = Delivery status for demand d by transportation mode l from merging center m in time period t ; *binary*

$QO^{k,l,m,t}$ = Delivery quantity for order k by transportation mode l from merging center m in time period t

$QD^{d,l,m,t}$ = Delivery quantity for demand d by transportation mode l from merging center m in time period t

$AO^{k,l,m,t}$ = Arrival quantity for order k by transportation mode l from merging center m in time period t

Costs/Profits - No Change

Production/Inventory - No Changes

Objective Function :

Maximize Profit - No Changes

Subject to:

(1) *Profits and Costs Definition*

(1.1) *Change to :*

$$H^k = \sum_{i \in I, l \in L, m \in M, t \in T} oc^{i,k} \cdot x^{i,os^k} \cdot QO^{k,l,m,t} \text{ for all } k \in K$$

Order profits are dependent on the particular configuration of the order, the profit margin and the quantity of the order that was delivered during the model timeframe

(1.2) *No Change :*

$$E^d = \sum_{i \in I} dc^{i,d} \cdot x^{i,ds^d} \cdot dq^d \cdot D^d \text{ for all } d \in D$$

Demand profits are dependent on the particular configuration of the demand, the profit margin and quantity reserved

(1.3) *Change to :*

$$CO^k = \sum_{l \in L, m \in M, t \in T} \left(u^l \cdot (1/ao) \cdot QO^{k,l,m,t} + v^l \cdot QO^{k,l,m,t} \right) \text{ for all } k \in K$$

Order delivery costs are based on the fixed costs of each committed order as well as the variable costs related to order quantity

(1.4) *Change to :*

$$CD^d = \sum_{l \in L, m \in M, t \in T} \left(u^l \cdot (1/ao) \cdot QD^{d,l,m,t} + v^l \cdot QD^{d,l,m,t} \right) \text{ for all } d \in D$$

Demand delivery costs are based on the fixed costs as well as the variable costs related to each delivery quantity

(1.5) *Change to :*

$$DD^k = \sum_{t=sl^{os^k}+1}^t \left(\left(t - sl^{os^k} \right) \cdot \sum_{l \in L, m \in M} AO^{k,l,m,t} \right) + \left(t + \text{Max}(sm^l) - sl^{os^k} \right) \cdot \left(oq^k - \sum_{l \in L, m \in M, t \in T} AO^{k,l,m,t} \right) \text{ for all } k \in K$$

Due date violation is the number of days past the requested service level due date that that an order arrives, including order quantities that are not delivered at all within the model timeframe

(2) *Order Delivery*

(2.1) *Change to :*

$$\sum_{l \in L, m \in M, t \in T} LO^{k,l,m,t} \leq y \text{ for all } k \in K$$

Orders must be delivered within allowable number of delivery splits

(2.2) *Change to :*

$$QO^{k,l,m,t} \leq oq^k \cdot LO^{k,l,m,t} \text{ for all } k \in K, l \in L, m \in M, t \in T$$

The delivery quantity each day (by each method) must be less than the requested amount

(2.3) *Change to :*

$$LO^{k,l,m,t} \leq QO^{k,l,m,t} \text{ for all } k \in K, l \in L, m \in M, t \in T$$

An order status is considered delivered by a certain transportation method only if an actual quantity is delivered to the customer

(2.4) *Change to :*

$$\sum_{l \in L, m \in M, t \in T} QO^{k,l,m,t} \leq oq^k \text{ for all } k \in K$$

The total amount delivered cannot be more than the requested amount

(2.5) *Change to :*

$$QO^{k,l,m,t} = AO^{k,l,m,t+sm^l+ll^{ol^k,m}} \text{ for all } k \in K, l \in L, m \in M, t \in T$$

The arrival of an order is dependent on the ship date of the order plus the delivery lead time of the shipping mode plus the lead time from the merging center to the customer region

(2.6) *Change to :*

$$AO^{k,l,t} = 0 \text{ for all } k \in K, l \in L, m \in M, t \in T \mid t \leq sm^l + ll^{ol^k,m}$$

The arrival of an order cannot occur in the beginning of the model, within the lead time for the specified delivery method

(3) *Demand Delivery*

(3.1) *Change to :*

$$\sum_{l \in L, m \in M, t \in T} LD^{d,l,m,t} \leq 1 \text{ for all } d \in D$$

Demand must be delivered in the same shipment (no splits)

(3.2) *Change to :*

$$\sum_{l \in L, m \in M, t \in T} (t + sm^l + ll^{dl^d,m}) \cdot LD^{d,l,m,t} \leq sl^d \text{ for all } d \in D$$

Demand must be delivered within allowable service level date

(3.3) *Change to :*

$$QD^{d,l,m,t} \leq dq^d \cdot LD^{d,l,m,t} \text{ for all } d \in D, l \in L, m \in M, t \in T$$

The delivery quantity cannot be more than the requested amount

(3.4) *Change to :*

$$D^d \leq \sum_{l \in L, m \in M, t \in T} LD^{d,l,m,t} \text{ for all } d \in D$$

Demand must be delivered if resources are reserved

(3.5) *Change to :*

$$\sum_{l \in L, m \in M, t \in T} QD^{d,l,m,t} = dq^d \cdot D^d \text{ for all } d \in D$$

When resources are reserved for demand, the quantity delivered must equal the percent reserved of demand

(4) *Material Conservation*

(4.1)-(4.6): *No Change*

(4.7) *Change to :*

$$\begin{aligned} ZK^{m,i,t} = & ZK^{m,i,t-1} + \sum_{f \in F \setminus \{f \mid t - sf^{f,m} < t\}} N^{f,m,i,t-sf^{f,m}} - \sum_{l \in L, k \in K \mid ol^{m,k}=1} oc^{i,k} \cdot QO^{k,l,m,t} \\ & - \sum_{l \in L, d \in D \mid dl^{m,d}=1} dc^{i,d} \cdot QD^{d,l,m,t} \quad \text{for all } m \in M, i \in KI, t \in T \end{aligned}$$

Inventory of kitting SKUs at each merging center is the previous day's inventory plus the quantity transferred in from each factory (accounting for the transfer lead time), less the quantity shipped to customers

(4.8) *No Change*

(4.9) *Change to :*

$$\begin{aligned} ZM^{m,i,t} = & ZM^{m,i,t-1} + pm^{m,i,t} - \sum_{l \in L, k \in K \mid ol^{i,k}=1} oc^{i,k} \cdot QO^{k,l,m,t} \\ & - \sum_{l \in L, d \in D \mid dl^{i,d}=1} dc^{i,d} \cdot QD^{d,l,m,t} \quad \text{for all } m \in M, i \in MI, t \in T \end{aligned}$$

Inventory of merging SKUs at merging centers is the previous day's inventory combined with daily stock supply, less amount shipped for orders and demand

In the experimental run, incoming orders are categorized by their geographic regions: East, Midwest and West. The closest merging centers are A, B, and C, respectively. In cases where a Merging SKU order is shipped from a merging center further away from the customer than its geographically-closest merging center, an additional lead time needs to be added to the delivery schedule. The following diagram details this scenario.

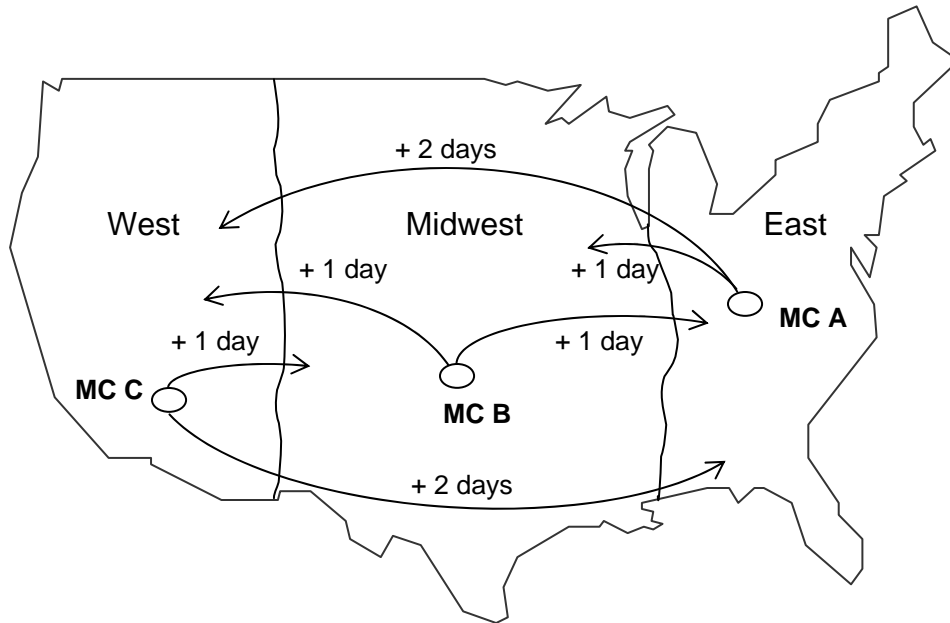


Figure 4.1: Lead Times from Merging Centers to Customer Regions

We ran 30 trials and compared the results to our base model.

Output	Base Scenario: Average Values	Exp. 3: No Affiliated Merging Center	Difference
Objective Function	1,225,632	1,235,565	0.8%
Profits	\$ 4,291,330	\$ 4,311,950	0.5%
Orders	\$ 2,444,865	\$ 2,448,917	0.2%
Demand	\$ 1,846,465	\$ 1,863,033	0.9%
Costs	\$ 1,787,663	\$ 1,799,045	0.6%
Order Delivery	\$ 996,839	\$ 999,143	0.2%
Due Date Violations	\$ 52,402	\$ 41,775	-20.3%
Demand Delivery	\$ 738,423	\$ 758,128	2.7%
Demand Commitment	73.0%	73.9%	1.3%
Gold Service Level	48.7%	46.2%	-5.2%
Silver Service Level	80.5%	81.8%	1.7%
Bronze Service Level	89.6%	93.5%	4.4%
kSKU1	87.0%	87.0%	0.0%
kSKU2	87.3%	87.8%	0.6%
kSKU3	83.0%	83.0%	0.0%
mSKU1	49.2%	50.3%	2.3%
mSKU2	58.4%	61.6%	5.5%

Table 4.15: Results of Experiment 3: Merging Center Re-Pointing

We can see that in general, the commitment levels and profits had insignificant changes with the new policy. Due date violation penalties, however, were down 20.3% from the base model. This would indicate that in cases where production capacity was limited, rather than manufacture an order late and deliver late, the order was shifted to another merging center. In addition, we can see that in the case of Merging SKUs, which are dependent only on inventory at the merging centers, reservations for demand increased by 2.3% and 5.5%. We can attribute this to the added flexibility in essentially transshipping Merging SKUs from one merging center to another to fulfill additional demand.

We next took a closer look at the results to study where the model re-pointed orders from one merging center to another. We studied in-depth a trial in which the data was identical for the base model and the new model and compared the results, as seen in the following tables.

Result Parameters	Base	Exp. 3	Diff.	Base	Exp. 3	Diff.
	Order Results			Demand Results		
kSKU1: Net	\$281,327	\$278,252	-1.1%	\$262,333	\$262,333	-0.1%
Profits	\$504,090	\$504,090	0.0%	\$455,747	\$455,747	0.0%
Del. Costs	\$215,998	\$215,998	0.0%	\$193,134	\$193,414	0.1%
DD Penalty	\$6,765	\$9,840	45.5%	-	-	-
kSKU2: Net	\$554,974	\$556,969	0.4%	\$410,344	\$413,119	0.7%
Profits	\$800,554	\$800,554	0.0%	\$584,675	\$588,822	0.7%
Del. Costs	\$233,760	\$232,560	-0.5%	\$174,331	\$175,702	0.8%
DD Penalty	\$11,820	\$11,025	-6.7%	-	-	-
kSKU3: Net	\$205,954	\$206,538	0.3%	\$166,529	\$165,251	-0.8%
Profits	\$428,194	\$428,194	0.0%	\$350,317	\$347,947	-0.7%
Del. Costs	\$215,519	\$218,115	1.2%	\$183,788	\$182,696	-0.6%
DD Penalty	\$6,720	\$3,540	-47.3%	-	-	-
mSKU1: Net	\$199,409	\$202,436	1.5%	\$153,172	\$162,739	6.2%
Profits	\$401,090	\$401,090	0.0%	\$258,764	\$290,287	12.2%
Del. Costs	\$195,261	\$195,309	0.0%	\$105,592	\$127,548	20.8%
DD Penalty	\$6,420	\$3,345	-47.9%	-	-	-
mSKU2: Net	\$219,543	\$224,083	2.1%	\$24,105	\$14,636	-39.3%
Profits	\$439,591	\$441,166	0.4%	\$51,640	\$32,884	-36.3%
Del. Costs	\$198,508	\$203,734	2.6%	\$27,535	\$18,248	-33.7%
DD Penalty	\$21,540	\$13,350	-38.0%	-	-	-

Table 4.16: Comparison of Results by SKU and Order/Demand

Result Parameters	Base	Exp. 3	Difference
ORDERS			
Total Net Profits	\$1,461,206	\$1,468,277	0.5%
Total Profits	\$2,573,518	\$2,575,094	0.1%
Total Del. Costs	\$1,059,047	\$1,065,716	0.6%
Total DD Penalty	\$53,265	\$41,100	-22.8%
DEMAND			
Total Net Profits	\$1,016,762	\$1,018,078	0.1%
Total Profits	\$1,701,143	\$1,715,686	0.9%
Total Del. Costs	\$684,381	\$697,607	1.9%
ORDERS + DEMAND			
Total Net Profits	\$2,477,969	\$2,486,355	0.3%
Total Profits	\$4,274,662	\$4,290,779	0.4%
Total Del. Costs	\$1,743,428	\$1,763,324	1.1%
Total DD Penalty	\$53,265	\$41,100	-22.8%

Table 4.17: Summarized Results of Merging Center Re-Pointing

Based on these results, we can conclude that the profits increase when the model can determine the merging center. Additionally, the new policy results in a sharp decline in due date penalty violation (22.8%).

The Kitting SKUs share both parts and production capacity, so the shifts are harder to pinpoint. The Merging SKUs however, have unique stock supplies, and can be studied with greater ease. We analyzed the 18 orders and forecast demand of mSKU1 to see where shifts occurred. The table below explains the major differences; any that are not mentioned do not shift merging centers, commitment levels or major shipping schedule differences (involving due date violations).

Demand/Order Details	Base Results	Exp. 3 Results
ID: Demand 29 Silver – Midwest 197 of mSKU1	Commitment: 0%	Commitment: 100% Ship from MC A (East)
ID: Demand 30 Bronze – Midwest 192 of mSKU1	Commitment: 0%	Commitment: 100% Ship from MC A (East)
ID: Demand 31 Gold – East 232 of mSKU1	Commitment: 100% Ship from MC A	Commitment: 55% Ship from MC A (East)
ID: Demand 34 Gold – West 309 of mSKU1	Commitment: 51% Ship from MC C	Commitment: 25.2% Ship from MC C (West)
ID: Order 28 Gold – Midwest 188 of mSKU1	Ship 188 from MC B \$480 in due date penalties	Ship 156 from MC B Ship 32 from MC C (West) \$0 in due date penalties
ID: Order 29 Silver - Midwest 224 of mSKU1	Ship 224 from MC B \$2,205 in due date penalties	Ship 176 from MC B Ship 48 from MC C (West) \$450 in due date penalties
ID: Order 30 Bronze – Midwest 189 of mSKU1	Ship 189 from MC B \$3,600 in due date penalties	Ship 189 from MC B \$1,095 in due date penalties
ID: Order 32 Silver – East 224 of mSKU1	Ship 224 from MC A \$0 in due date penalties	Ship 224 from MC A \$1,665 in due date penalties

Table 4.18: Effect of Merging Center Re-pointing on Orders/Demand

From this table, we can see that the stock levels of mSKU1 were scarcest in the Midwest region. Orders in the base trial have high associated due date penalties from this merging center (\$480, \$2,205, and \$3600, for each respective service level). Additionally, the forecast demand for this region is not committed at all for the Silver or Bronze service levels.

However, when we alter the policy so that orders/demand can be fulfilled from any merging center, we can see that shifts have been made to alleviate the shortages in the Midwest. The due date penalties for orders in the Midwest region have dramatically decreased (to \$0, \$450, and \$1,095, respectively). Additionally, more resources could be reserved from other merging centers to fulfill higher commitment of demand – up to 100% for the Silver and Bronze service levels.

In some cases, the other two merging centers may have had surplus inventory of mSKU1 and could fulfill the extra demand without issue. However, if the inventory levels at the other merging centers were more limited, than the model must make choices to determine which demand should be fulfilled and which orders should be shifted to later deliveries to fulfill higher profit margin demand. The following three graphs illustrate how resources were shifted to fulfill additional demand and schedule orders more efficiently.

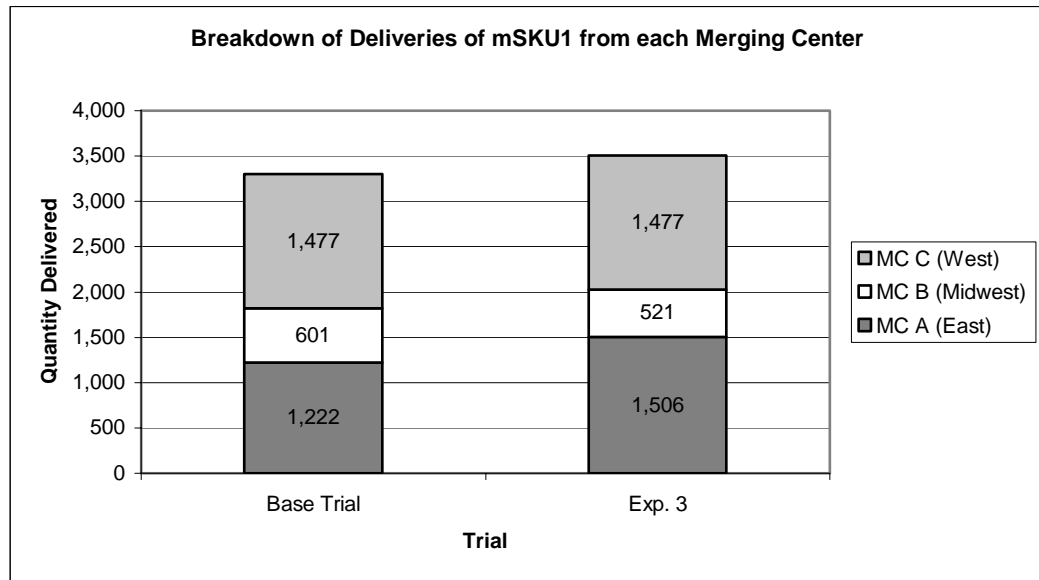


Figure 4.2: Chart of Delivery Quantities from each Merging Center

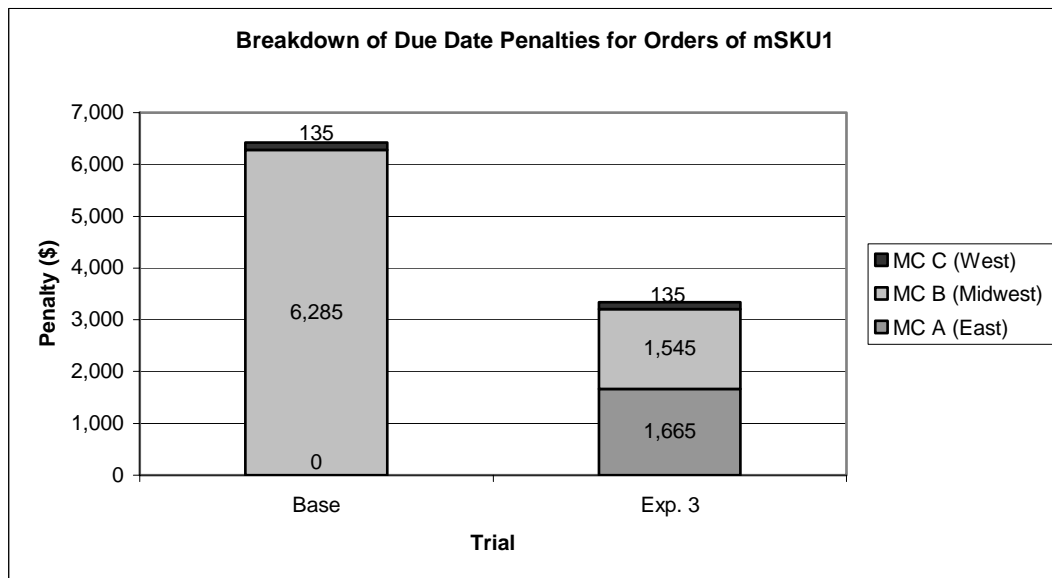


Figure 4.3: Chart of Due Date Penalties per Merging Center

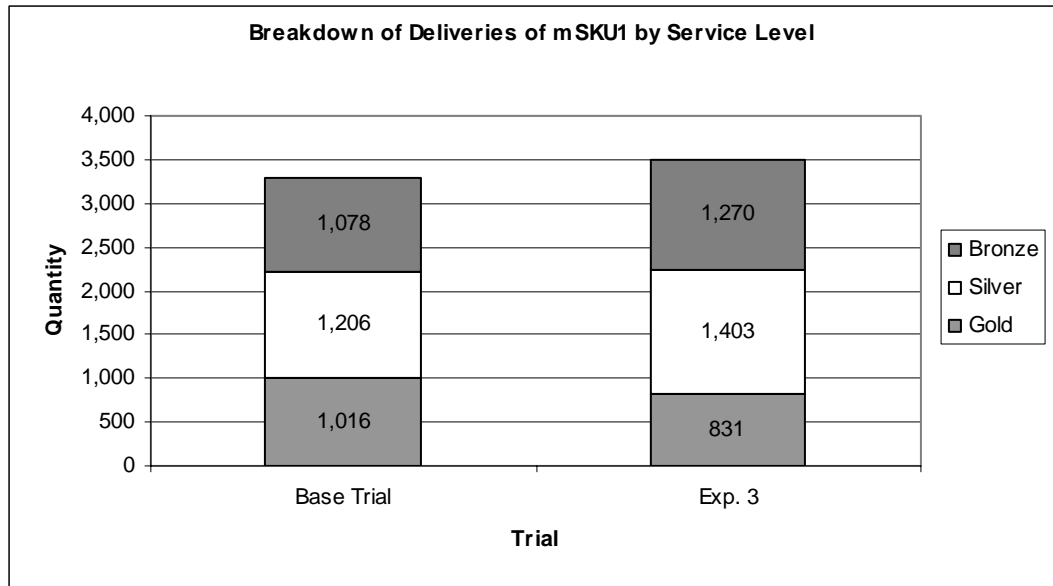


Figure 4.4: Chart of Delivery Quantities per Service Level

Contrary to our initial belief, the quantity of Gold service level orders/demand for mSKU1 that are delivered does not increase with the new policy. We would have expected that the model would seek to reserve resources for the higher profit margin service level. However, if we analyze the results, we can see that the overall commitment of mSKU1 increased by 204 units, which is comprised of large increases of Bronze and Silver commitment levels, yet a decrease in the Gold service level commitment. We gather that the additional profits from the Silver and Bronze commitment outweigh the profits lost with the decrease of Gold commitment.

4.6. Experiment 4: Re-pointing of Production

Capacity

In another kind of re-pointing, an order can be shifted from one factory to another for production, or from a scheduled production date to one at a later date. By re-assigning orders, the model can reserve that capacity for higher profit demand. We chose to analyze the scenario in which the production capacity is reduced for one of the factories. In the base model, there were two factories, Factory A in the East, and Factory B in the Midwest. We reduce the production capacity at Factory A by 30% and check the results against the base model.

We expect to see the capability of the model to shift production of orders from Factory A to Factory B. Additionally, some of the orders originally produced at Factory A will be pushed to later delivery dates (for a penalty) and some of the demand will shift to uncommitted status. We expect that these decisions will be made based on higher profit margins (based on SKU or service level).

We ran the experiment over 10 trials and compared the results to the base run. The results are presented in the following table.

Output	Base Scenario: Average Values	Exp. 4: Production Re-Pointing	Difference
Objective Function	1,228,620	1,176,393	-4.4%
Profits	\$4,283,091	\$4,147,757	-3.3%
Orders	\$2,417,190	\$2,417,190	0.0%
Demand	\$1,865,900	\$1,730,567	-7.8%
Costs	\$1,777,693	\$1,734,301	-2.5%
Order Delivery	\$981,731	\$990,210	0.9%
Due Date Violations	\$48,158	\$60,671	20.6%
Demand Delivery	\$747,804	\$683,420	-9.4%
Demand Commitment	73.3%	67.3%	-9.0%
Gold Service Level	48.5%	38.2%	-27.0%
Silver Service Level	80.1%	72.4%	-10.7%
Bronze Service Level	91.2%	91.2%	0.0%
kSKU1	87.0%	86.7%	-0.4%
kSKU2	87.4%	87.4%	0.0%
kSKU3	84.2%	54.0%	-55.9%
mSKU1	48.9%	48.9%	0.0%
mSKU2	56.9%	56.9%	0.0%

Table 4.19: Results of Experiment 4: Production Center Re-pointing

The due date violations increased, which is reasonable considering that production capacity was limited in the re-pointing experiment. There is a drastic decrease in the commitment of orders of Kitting SKU3, by 55.9%. We expect this is due to the lower profit margin associated with this Kitting SKU. To verify, we compared the results of a single trial from the base run with the experimental run to analyze the specific shifts made. The results of this trial run are presented in the following table.

Output	Base Scenario: Average Values	Exp. 4: Production Re-Pointing	Difference
Objective Function	1,284,321	1,248,867	-2.8%
Profits	\$4,504,115	\$4,378,511	-2.9%
Orders	\$2,442,899	\$2,442,899	0.0%
Demand	\$2,061,216	\$1,935,612	-6.5%
Costs	\$1,887,489	\$1,827,121	-3.3%
Order Delivery	\$1,016,623	\$1,012,910	-0.4%
Due Date Violations	\$47,985	\$53,655	10.6%
Demand Delivery	\$822,881	\$760,557	-8.2%
Demand Commitment	82.5%	76.8%	-7.4%
Gold Service Level	60.7%	52.8%	-15.0%
Silver Service Level	88.6%	80.1%	-10.6%
Bronze Service Level	100.0%	100.0%	0.0%
kSKU1	86.6%	86.6%	0.0%
kSKU2	86.7%	86.7%	0.0%
kSKU3	77.6%	48.7%	-59.2%
mSKU1	71.6%	71.6%	0.0%
mSKU2	89.4%	89.4%	0.0%

Table 4.20: Comparison of Results of Production Re-pointing for Single Trial (Same Data)

To accurately study how decisions are made, we must determine the descending order of products based on profit margins. We ignore mSKU1 and mSKU2, as they are not affected by the production shortage at Factory A. The following table displays the sorted Kitting SKUs.

Parameter	Profit Margin
kSKU2: Gold	414.69
kSKU2: Silver	369.42
kSKU2: Bronze	334.91
kSKU1: Gold	285.09
kSKU1: Silver	252.78
kSKU3: Gold	237.09
kSKU1: Bronze	226.91
kSKU3: Silver	209.58
kSKU3: Bronze	186.91

Table 4.21: Kitting SKUs Sorted by Profit Margin

The profit margin of Kitting SKU2 is higher than all other products, for any service level. Kitting SKU1 has the next highest profit margin. Thus, we expect that capacity will be reserved first for kSKU2. In looking at the results, we can see that this is true. We compared the difference in revenues (profits less costs and due date penalties) for the orders and demand of each product and service level. The following chart displays the results.

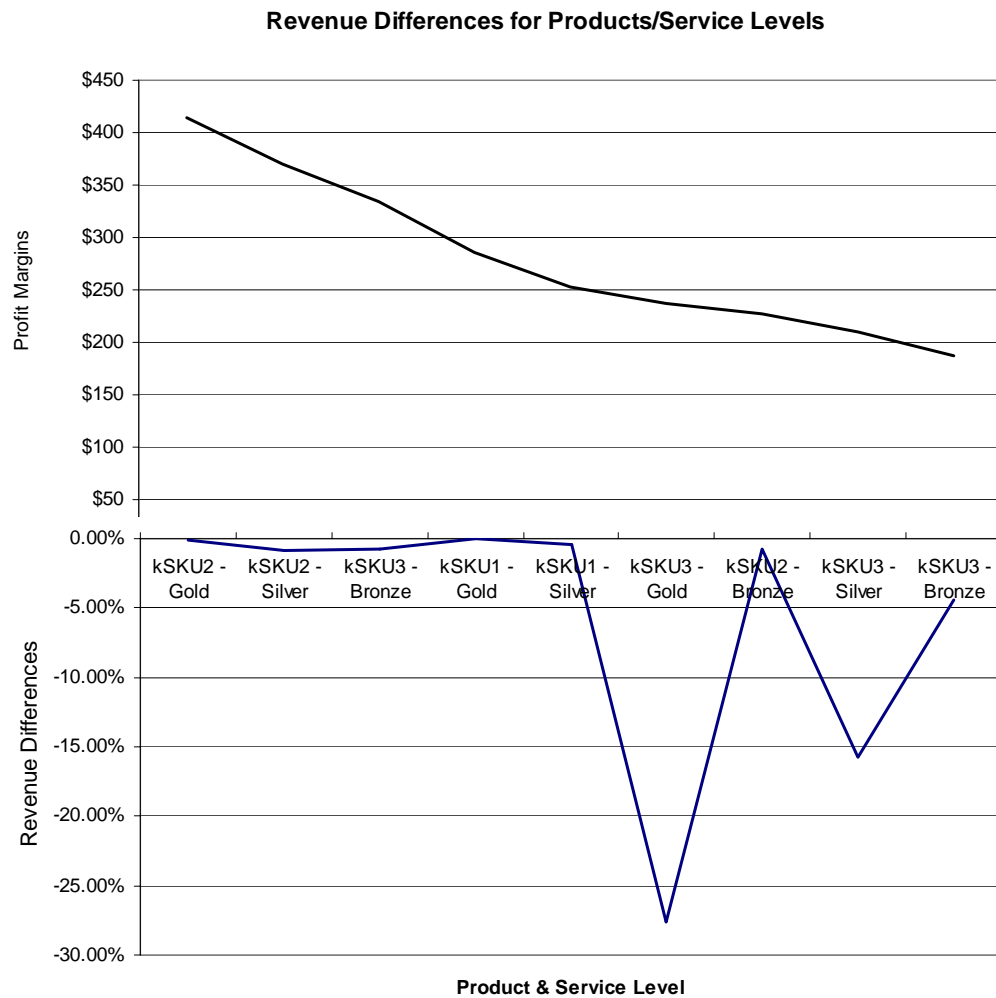


Figure 4.5: Chart of Revenue Differences across the Product/Service Level Configurations

We can see that the production is reserved first for the highest profit margin products. The product-service level configurations with the smallest profit margins have the greatest differences in revenue. Thus, the model has re-pointed capacity that was reserved for those lower profit margin orders/demand in the base run during this experiment. The capacity is instead reserved for the higher profit margin products.

5. Conclusion

In this thesis, we described a mixed integer programming model that integrates the order assignment function and the resource scheduling function for an assemble-to-order environment. The model determines the optimal production schedule based on accepted orders and forecast demand. It trades off resource reservation for demand and order delivery schedules based on profitability considerations.

5.1. Summary of Results

Our experiments proved the capability of the model to effectively trade off lower profit margin accepted orders with uncertain, higher profit future demand. The sensitivity analysis highlighted this capability in different scenarios. When the profit margin of a particular configuration of SKU and service level is increased, the model re-allocates resources to commit this demand first. Additionally, in the case where capacity was tight, the model effectively reserved demand for the higher profit margin products over the lower profit margin products.

In our first experiment, we tested the policy of commitment level. We showed that the manufacturer can increase revenues if it is able to reserve a portion of the aggregate demand for each configuration, rather than an all-or-nothing commitment policy.

Next, we proved that our policy for service levels is effective in maximizing revenue. When the model chooses the delivery mode and schedule for orders (instead of the customer), the resource allocation becomes more flexible. This is successful in increasing the commitment levels and reducing the due date violations, resulting in additional revenue over the standard policy.

Finally, we showed that the policy in which orders are not assigned specific merging centers is useful. Instead of aligning orders with merging centers based on geographic proximity alone, the model also considers capacity allocation. This results in fewer uncommitted demand orders, and thus maximizes overall revenue.

The experimental results also proved the usefulness of the spreadsheet-based front end of the model. We could quickly update any data parameter values and analyze the results of the model optimization with ease.

5.2. Future Work

There are several possible extensions of our research. The most significant impact would be to enhance the model by considering a rolling execution mode. In its current state, the model only considers accepted orders and demand for a single run. The rolling timeframe setup, however, would consider the previously promised orders and demand in setting the resource levels, providing a much more accurate depiction of the available resource capabilities.

It would also be interesting to extend the model formulation to include the concept of *carry-over demand*. This is defined as the percentage of uncommitted demand for a higher service level that will shift to a lower service level. For

instance, if a company cannot reserve resources for orders of a particular SKU at the Gold service level, a decent percentage of those customers would find a lower service level (Silver) acceptable. The model currently is formulated in a manner that does not account for any of this carry-over demand, which does not give an accurate representation of real-life.

Additional policies relating to order promising and resource booking can be analyzed using this model. An interesting policy study would be to analyze both the customer channel and profit margin of forecast demand when deciding whether to reserve capacity. For instance, an order from a loyal corporate client might be given more weight than an order of the same configuration from a new client. Not all ATO/CTO firms will be able to differentiate customers, but when applicable, this added feature would be beneficial in analyzing fully the intricacies involving demand reservation.

Future research may also study the costing and pricing mechanisms of the model, in respect to the sales and marketing functions of the manufacturing. Experimentation can be conducted to determine if a new pricing scheme is effective using our model. For instance, if the marketing team wanted to run a promotion offering a free service level upgrade for certain under-utilized product configurations, the model can be used a decision support system to determine the effects on revenue and commitment levels of this proposed policy.

Appendix A: Xpress Mosel Code

The Mosel code used in the base setup of the model is provided below.

```
model "Order Assignment and Resource Reservation"
uses  "mmodbc", "mmxprs"

declarations
    SQLStr: string
end-declarations

setparam("XPRS_VERBOSE", true)
setparam("XPRS_LOADNAMES", true)

! Connect to the Excel spreadsheet
SQLStr := 'DSN=[Name]; DBQ=[Name].xls'

declarations
    NUM = 1..1          ! For arrays with just one value
    NT = 10
    T = 1..NT           ! Time periods
    Tp1 = 0..NT         ! t+1 time period
    TL = 1..13          ! Time periods, plus lead time for
                        ! delivery from last day of production
    avgSize = 10        ! Average SKUs per order is 10

!Index Parameters
coID: set of string      ! Customer orders
dmdID: set of string     ! Forecast demand
serLvls: set of string   ! Service levels
transModes: set of string ! The transportation modes
kSKUs: set of string     ! SKUs that need kitting from parts
mSKUs: set of string     ! SKUs that only need merging
kParts: set of string    ! The kitting parts
mCenters: set of string  ! The merging centers
Factories: set of string ! The factories for kitting
SKUs: set of string      ! All SKUs = mSKUs + kSKUs

!Order Parameters
maxDelTimes: array(1..1) of real ! Number of times orders can be
                                ! split for delivery
ordCmit: array(coID) of integer  ! Commitment status of orders
ordQty: array(coID) of integer   ! Order quantity
ordSerLvl: array(coID) of string ! Order service level
ordCfg: array(SKUs,coID) of integer ! Order configuration
ordLoc: array(mCenters, coID) of integer ! Order location
```

```

!Demand Parameters
dmdQty: array(dmdID) of integer          ! Demand quantity
dmdSerLvl: array(dmdID) of string        ! Demand service level
dmdCfg: array(SKUs,dmdID) of integer     ! Demand configuration
dmdLoc: array(mCenters, dmdID) of integer ! Demand location

!Cost Parameters
wgtProfit: array(1..1) of real           ! Weight of profits in obj.
wgtCost: array(1..1) of real             ! Weight of costs in obj.
wgtDueDate: array(1..1) of real          ! Weight of due date violation
                                           in obj.
tFixCost: array(transModes) of real      ! Fixed transportation costs
tVarCost: array(transModes) of real      ! Variable transportation costs
profitMgn: array(SKUs,serLvls) of real   ! Unit profit margin of
                                           each SKU under each service level

!Production/Merging/Delivery Parameters
bom: array(kSKUs,kParts) of integer       ! The bill of materials
                                           for SKUs
prodCap: array(Factories,T) of integer    ! Production capacity
prodLT: array(Factories,mCenters) of integer ! Production lead
                                           time from factory to merging centers
tLeadTime: array(transModes) of integer   ! Trans. lead times
serDays: array(serLvls) of integer        ! Service level timeframe

!Inventory Parameters
KPAvil: array(Factories,kParts, T) of integer ! Kitting part stock
initPartInv: array(Factories,kParts) of integer ! Initial
                                           inventory of parts at each factory
initKSKUF: array(Factories, kSKUs) of integer ! Initial inventory
                                           of kSKUs at each factory
initKSKUM: array(mCenters,kSKUs) of integer ! Initial inventory
                                           of kSKUs at merging center
initMSKU: array(mCenters,mSKUs) of integer ! Initial inventory
                                           of mSKUs at merging center
mSKUAvil: array(mCenters, mSKUs, T) of integer ! Availability of
                                           merging parts

end-declarations

setparam("SQLIdxcol", true)              ! Index reference values

SQLconnect(SQLStr)
SQLexecute("SELECT * FROM serLvlRng", serLvls)
SQLexecute("SELECT * FROM mCenterRng", mCenters)
SQLexecute("SELECT * FROM facRng", Factories)
SQLexecute("SELECT * FROM kSKURng", kSKUs)
SQLexecute("SELECT * FROM mSKURng", mSKUs)
SQLexecute("SELECT * FROM kPartRng", kParts)
SQLexecute("SELECT * FROM transModeRng", transModes)
SQLexecute("SELECT * FROM ordIDRng", coID)
SQLexecute("SELECT * FROM dmdIDRng", dmdID)
SQLdisconnect

```

```

writeln(serLvls); writeln(mCenters); writeln(Factories);
writeln(mSKUs); writeln(kParts); writeln(transModes);
writeln(mCenters); writeln(kSKUs)
!writeln(dmdID)

finalize(coID); finalize(serLvls); finalize(transModes);
finalize(dmdID); finalize(kSKUs); finalize(mSKUs); finalize(kParts);
finalize(mCenters); finalize(Factories)

SKUs:= kSKUs + mSKUs      ! Total SKUs for both kitting and merging
finalize(SKUs)
!writeln(SKUs);

setparam("SQLndxcol", false)
SQLconnect(SQLStr)
SQLexecute("SELECT * FROM serDaysRng", [serDays])
SQLexecute("SELECT * FROM bomRng", [bom])
SQLexecute("SELECT * FROM tFixCostRng", [tFixCost])
SQLexecute("SELECT * FROM tVarCostRng", [tVarCost])
SQLexecute("SELECT * FROM transLTRng", [tLeadTime])
SQLexecute("SELECT * FROM prodLTRng", [prodLT])
SQLexecute("SELECT * FROM ordQtyRng", [ordQty])
SQLexecute("SELECT * FROM ordCfgRng", [ordCfg])
SQLexecute("SELECT * FROM ordLocRng", [ordLoc])
SQLexecute("SELECT * FROM ordSerLvlRng", [ordSerLvl])
SQLexecute("SELECT * FROM dmdQtyRng", [dmdQty])
SQLexecute("SELECT * FROM dmdCfgRng", [dmdCfg])
SQLexecute("SELECT * FROM dmdLocRng", [dmdLoc])
SQLexecute("SELECT * FROM dmdSerLvlRng", [dmdSerLvl])
SQLexecute("SELECT * FROM mSKUAvilRng", [mSKUAvil])
SQLexecute("SELECT * FROM partAvilRng", [KPAvil])
SQLexecute("SELECT * FROM prodCapRng", [prodCap])
SQLexecute("SELECT * FROM initMSKURng", [initMSKU])
SQLexecute("SELECT * FROM initPartRng", [initPartInv])
SQLexecute("SELECT * FROM profitMgnRng", [profitMgn])
SQLexecute("SELECT * FROM wgtCostRng", [wgtCost])
SQLexecute("SELECT * FROM wgtProfitRng", [wgtProfit])
SQLexecute("SELECT * FROM wgtDueDateRng", [wgtDueDate])
SQLexecute("SELECT * FROM maxDelTimesRng", [maxDelTimes])
SQLexecute("SELECT * FROM initKSKUFRng", [initKSKUF])
SQLexecute("SELECT * FROM initKSKUMRng", [initKSKUM])
SQLdisconnect

!writeln(serDays); writeln(bom); writeln(tFixCost);
writeln(tVarCost); writeln(tLeadTime); writeln(prodLT);
writeln(ordQty); writeln(ordCfg); writeln(ordLoc);
writeln(ordSerLvl); writeln(ordCmit); writeln(Dmd);
writeln(mSKUAvil); writeln(KPAvil); writeln(prodCap);
writeln(mergeCap); writeln(initMSKU); writeln(initPartInv);
writeln(profitMgn); writeln(wgtCost); writeln(wgtProfit);
writeln(maxDelTimes); writeln(mergeCost); writeln(prodCost);
writeln(initKSKUF); writeln(initKSKUM); writeln(dmdQty);
writeln(dmdCfg); writeln(dmdLoc); writeln(dmdSerLvl)

```

```

declarations

! Order/demand decision variables

LT_SET: array(dmdID) of mpvar
! Commitment % of demand

ORD_DEL: array(coID,transModes,T) of mpvar
! Delivery status of order by each trans. mode (0,1)

DMD_DEL: array(dmdID,transModes,T) of mpvar
! Delivery status of demand by each trans. mode (0,1)

ORD_DQTY: array(coID,transModes,T) of mpvar
! Delivery quantity of order

DMD_DQTY: array(dmdID,transModes,T) of mpvar
! Delivery quantity of demand

ORD_AQTY: array(coID, transModes,TL) of mpvar
! Arrival quantity of order

! Cost/profit decision variables

ORD_PFT: array(coID) of mpvar
! Profit from each order

DMD_PFT: array(dmdID) of mpvar
! Profit from each demand

ORD_COST: array(coID) of mpvar
! Cost for order delivery

DMD_COST: array(dmdID) of mpvar
! Cost for demand delivery

DD_VIOLATION: array(coID) of mpvar
! Due date violation costs

! Production/inventory decision variables

PROD_QTY: array(Factories,kSKUs,T) of mpvar
! Production quantity

TRANS_QTY: array(Factories, mCenters, kSKUs, T) of mpvar
! Quantity of kSKUs transferred from factories to merging centers

TRANS_ARR_QTY: array(Factories, mCenters, kSKUs, T) of mpvar
! Arrival quantity of kSKUs

PART_INV: array(Factories, kParts, Tpl) of mpvar
! Inventory level of parts at factories

```

```

M_INV: array(mCenters,mSKUs,Tp1) of mpvar
! Inventory level of mSKUs at merging centers

K_INV_FAC: array(Factories, kSKUs, Tp1) of mpvar
! Inventory level of kSKUs at factories

K_INV_MC: array(mCenters, kSKUs, Tp1) of mpvar
! Inventory level of kSKUs at merging centers

end-declarations

! Stop branch and bound when within certain level of best bound
setparam("XPRS_MIPABSSTOP", 5)

! Objective Function-----
PROFIT := wgtProfit(1) * sum(k in coID) ORD_PFT(k) +
          wgtProfit(1) * sum(d in dmdID) DMD_PFT(d) -
          wgtCost(1) * (sum(k in coID) ORD_COST(k) + sum(d in dmdID)
          DMD_COST(d)) - wgtDueDate(1) * sum(k in coID) DD_VIOLATION(k)

! Constraints-----
! Profits and Cost Definition

!1.1 Order profits depend on configuration, profit margin and
delivered quantity
forall(k in coID)
    ORD_PFT(k) = sum(i in SKUs, l in transModes, t in T)
                ordCfg(i,k) * profitMgn(i,ordSerLvl(k))*
                ORD_DQTY(k,l,t)

!1.2 Demand profits depend on SKUs, profit margin and quantity for
committed demand
forall(d in dmdID)
    DMD_PFT(d) = sum(i in SKUs)
                dmdCfg(i,d) * profitMgn(i,dmdSerLvl(d)) * dmdQty(d) *
                LT_SET(d)

!1.3 Order costs depend on fixed and variable costs of
transportation method
forall (k in coID)
    ORD_COST(k) = sum(l in transModes, t in T)
                  (tFixCost(1)*(1/avgSize)*ORD_DQTY(k,l,t) +
                  tVarCost(1)*ORD_DQTY(k,l,t))

!1.4 Demand costs depend on fixed and variable costs of
transportation method
forall (d in dmdID)
    DMD_COST(d) = sum(l in transModes, t in T)
                  (tFixCost(1)*(1/avgSize)*DMD_DQTY(d,l,t) +
                  tVarCost(1)*DMD_DQTY(d,l,t))

```

```

!1.5 Due date violation for orders is the days late the order is,
by the quantity late, plus the portion of an order not delivered
within the timeframe
forall (k in coID)
    DD_VIOLATION(k) = sum(t in TL | t >= (serDays(ordSerLvl(k))))
        ((t-serDays(ordSerLvl(k)))*sum(l in transModes)
        ORD_AQTY(k,l,t)) + (13-serDays(ordSerLvl(k))) *
        (ordQty(k) - sum(t in T, l in transModes)
        ORD_AQTY(k,l,t))

! Order Delivery Definition
!2.1 Orders must be delivered within maximum number of times
forall(k in coID)
    sum(l in transModes, t in T) ORD_DEL(k,l,t) <= sum(n in NUM)
        maxDelTimes(n)

!2.2 Delivery quantity must be less than order commitment quantity
forall(k in coID, l in transModes, t in T)
    ORD_DQTY(k,l,t) <= ordQty(k) * ORD_DEL(k,l,t)

!2.3 Delivery status = 1 only if actual quantity is delivered
forall(k in coID, l in transModes, t in T)
    ORD_DEL(k,l,t) <= ORD_DQTY(k,l,t)

!2.4 Total amount of order delivered must be less than the
requested amount
forall(k in coID)
    sum(l in transModes, t in T) ORD_DQTY(k,l,t) <= ordQty(k)

!2.5 Define the arrival day of order to customer
forall(k in coID, l in transModes, t in T)
    ORD_DQTY(k,l,t) = ORD_AQTY(k,l,t+tLeadTime(l))

!2.6 Can't have any arrival qty, unless delivered
forall(k in coID, l in transModes, t in TL | t <= tLeadTime(l))
    ORD_AQTY(k,l,t) = 0

! Demand Delivery Definition
!3.1 Demand must be delivered at same time (no splitting)
forall(d in dmdID)
    sum(l in transModes, t in T) DMD_DEL(d,l,t) <= 1

!3.2 Demand must be delivered within due date (service level)
forall(d in dmdID)
    sum(l in transModes, t in T) (t + tLeadTime(l))*DMD_DEL(d,l,t)
        <= serDays(dmdSerLvl(d))

!3.3 Daily delivery quantity must be less than total requested
amount
forall(d in dmdID, l in transModes, t in T)
    DMD_DQTY(d,l,t) <= dmdQty(d) * DMD_DEL(d,l,t)

!3.4 Demand must be delivered if it is committed
forall(d in dmdID)

```



```

LT_SET(d) <= sum(l in transModes, t in T) DMD_DEL(d,l,t)

!3.5 Total amount delivered must equal the percent of demand
committed
forall(d in dmdID)
    sum(l in transModes, t in T) DMD_DQTY(d,l,t) = dmdQty(d) *
        LT_SET(d)

! Material Conservation
!4.1 Initial inventory of kitting parts at factory
forall(f in Factories, j in kParts)
    PART_INV(f,j,0)=initPartInv(f,j)

!4.2 Flow of kitting parts at factory
forall(f in Factories, j in kParts, t in T)
    PART_INV(f,j,t) = PART_INV(f,j,t-1) + KPAvil(f,j,t) - sum(i in
        kSKUs) bom(i,j)*PROD_QTY(f,i,t)

!4.3 Production must be within capacity for each factory
forall(f in Factories, t in T)
    sum(i in kSKUs) PROD_QTY(f,i,t) <= prodCap(f,t)

!4.4 Initial inventory of kitting SKUs at each factory
forall(f in Factories, i in kSKUs)
    K_INV_FAC(f,i,0) = initKSKUF(f,i)

!4.5 Flow of kitting SKUs at factory
forall(f in Factories, i in kSKUs, t in T)
    K_INV_FAC(f,i,t) = K_INV_FAC(f,i,t-1) + PROD_QTY(f,i,t) - sum(m
        in mCenters) TRANS_QTY(f,m,i,t)

!4.6 Initial inventory kitting SKUs at each merging center
forall(m in mCenters, i in kSKUs)
    K_INV_MC(m,i,0) = initKSKUM(m,i)

!4.7 Flow of kitting SKUs at merging center
forall(m in mCenters, i in kSKUs, t in T)
    K_INV_MC(m,i,t) = K_INV_MC(m,i,t-1) +
        sum(f in Factories | prodLT(f,m)<t) TRANS_QTY(f,m,i,t-
            prodLT(f,m)) - sum(l in transModes, k in coID |
            ordLoc(m,k)=1) ordCfg(i,k)*ORD_DQTY(k,l,t) -
            sum(l in transModes, d in dmdID | dmdLoc(m,d)=1)
            dmdCfg(i,d)*DMD_DQTY(d,l,t)

!4.8 Initial inventory of merging SKUs at each merging center
forall(m in mCenters, i in mSKUs)
    M_INV(m,i,0) = initMSKU(m,i)

!4.9 Flow of merging SKUs at merging center
forall(m in mCenters, i in mSKUs, t in T)
    M_INV(m,i,t) = M_INV(m,i,t-1) + mSKUAvil(m,i,t) -
        sum(k in coID, l in transModes | ordLoc(m,k)=1)
        ordCfg(i,k)* ORD_DQTY(k,l,t)- sum(l in transModes, d in
            dmdID | dmdLoc(m,d)=1) dmdCfg(i,d)*DMD_DQTY(d,l,t)

```

```

! Boundary Constraints-----
! Define all variables as either continuous or binary
forall(d in dmdID) DMD_COST(d) is_continuous
forall(k in coID) ORD_COST(k) is_continuous
forall(d in dmdID) LT_SET(d) <=1
forall(d in dmdID) LT_SET(d) is_continuous
forall(d in dmdID, l in transModes, t in T) DMD_DEL(d,l,t)
    is_binary
forall(k in coID, l in transModes, t in T) ORD_DEL(k,l,t)
    is_binary
forall(d in dmdID, l in transModes, t in T) DMD_DQTY(d, l, t)
    is_continuous
forall(k in coID, l in transModes, t in T) ORD_DQTY(k,l,t)
    is_continuous
forall(f in Factories, j in kParts, t in Tp1) PART_INV(f,j,t)
    is_continuous
forall(f in Factories, i in kSKUs, t in Tp1) K_INV_FAC(f,i,t)
    is_continuous
forall(m in mCenters, i in kSKUs, t in Tp1) K_INV_MC(m,i,t)
    is_continuous
forall(m in mCenters, i in mSKUs, t in Tp1) M_INV(m,i,t)
    is_continuous
forall(k in coID) ORD_PFT(k) is_continuous
forall(d in dmdID) DMD_PFT(d) is_continuous
forall(f in Factories, i in kSKUs, t in T) PROD_QTY(f,i,t)
    is_continuous
forall(f in Factories, m in mCenters, i in kSKUs, t in T)
    TRANS_QTY(f,m,i,t) is_continuous
forall(k in coID) DD_VIOLATION(k) is_continuous

! Solve the problem
maximize(PROFIT)

! Give solution values in Xpress:
writeln("Obj:=", getobjval)

forall (k in coID) writeln ("Due Date Violation:=(",k,") = ",
    getsol(DD_VIOLATION(k)))

!forall (d in dmdID) writeln("Demand Costs(",d,") =",
    getsol(DMD_COST(d)))
d-do
!)

end-model

```

Appendix B: Selected Excel VB Code

Several modules were created in Visual Basic for the model in Excel. Some of the more interesting/complex code has been included here.

Subroutine to call Xpress solver and import results:

```
Option Explicit

Public Sub runOA_LTS_Base()

Const ROOT = "C:\XpressMP\"
Const SOURCE_PATH = ROOT & "[filename].mos"
Const BIM_PATH = ROOT & "[filename].bim"
Const XLS_PATH = ROOT & "[filename].xls"

Dim nReturn As Integer
Dim model As Long

' Redirect the mosel stdout and stderr
XPRMsetStream XPRMIO_OUT, ROOT & "log.txt"
XPRMsetStream XPRMIO_ERR, ROOT & "err.txt"

' Initialize mosel
nReturn = XPRMinit
If XPRMinit() <> 0 Then
    MsgBox "Failed to initialize Mosel"
    Exit Sub
End If

' Compile model source file to binary .bim file
nReturn = XPRMcompmod("", SOURCE_PATH, BIM_PATH, "")
If nReturn <> 0 Then
    If nReturn = 1 Then
        MsgBox "Parsing phase has failed (syntax error or file access error)"

        Exit Sub

    ElseIf nReturn = 2 Then
        MsgBox "Error in compilation phase (detection of a semantic error)"
```

```

Exit Sub

ElseIf nReturn = 3 Then
    MsgBox "Error writing the output file"

Exit Sub

Else
    MsgBox "Failed to compile mosel file"

End If
End If

' Load the binary model into mosel
model = XPRMloadmod(BIM_PATH, "")

' Execute the model
Dim result As Long
nReturn = XPRMrunmod(model, result, "DATA_XLS='" & XLS_PATH & "'")
If nReturn <> 0 Then
    MsgBox "Error during execution of model"
Exit Sub
End If

' Get solution results
Dim i, f, j, k, l, t, s, d As Integer
Dim index() As Long
Dim handle As Variant
Dim mpvar As Variant

Const MAXDMD = 45
Const MAXORDS = 45
Const MAXTRANSMODE = 3
Const MAXTIMES = 10
{continue for other parameters}

' Get objective value
Dim objval As Double
objval = XPRMgetobjval(model)
Worksheets("Results Tab").Cells([#],[#]) = objval

' Get commitment value for demand
' Request a handle to the "LT_SET" mpvar array
' Iterate through the array, retrieving the values
Call XPRMfindident(model, "LT_SET", handle)
ReDim index(0)
For d = 1 To MAXDMD
    ' Indexing array
    index(0) = d

    ' Retrieve an mpvar element from the Mosel vars array
    Call XPRMgetarrval(handle, index, mpvar)

```

```

        ' Extract the solution value from the mpvar
        Worksheets("[Results Tab]").Cells([#], [#] + d - 1) =
        XPRMgetvsol(model, mpvar)
    Next

'Get order delivery quantity
Call XPRMfindident(model, "ORD_DQTY", handle)
    ReDim index(2)
    Dim count_k As Integer
    Dim count_t As Integer
    Dim tmpVal as Long
    count_t = 0
    count_k = 0
    For k = 1 To MAXORDS
        tmpVal = 0
        For l = 1 To MAXTRANSMODE
            For t = 1 To MAXTIMES
                index(0) = k: index(1) = l: index(2) = t
                Call XPRMgetarrval(handle, index, mpvar)
                tmpVal = XPRMgetvsol(model, mpvar)
                If tmpVal > .9 Then
                    Worksheets("[ResultsTab]").Cells([#]
                    + count_t, [#] + count_k) = tmpVal

                    count_t = count_t + 3

                End If
            Next t
        Next l
        count_k = count_k + 1
        count_t = 0
    Next k

{Continue in similar fashion importing all desired results}

End Sub

```

References

- Akkan, C., "Finite-Capacity Scheduling-Based Planning for Revenue-Based Capacity Management," *European Journal of Operational Research*, 100, 170-179, 1997.
- Andel, T., "Material Handling @ amazon.com," *Material Handling Management*, 55: 1, 38, 2000.
- Bacheldor, B., "From Scratch: Amazon Keeps Supply Chain Close to Home," *InformationWeek*, 979, 40, 2004.
- Balakrishnan, N., S.V. Sridharan, J.W. Patterson, "Rationing Capacity between Two Product Classes," *Decision Sciences*, 27: 2, 185-214, 1996.
- Barut, M., V. Sridharan, "Revenue Management in Order-Driven Production Systems," *Decision Sciences*, 36: 2, 287-316, 2005.
- Buderi, R., "Direct from Dell," *Technology Review*, 104: 6, 78-83, 2001.
- Butler, J.C., J.S. Dyer, "Optimizing Natural Gas Flows with Linear Programming and Scenario," *Decision Sciences*, 30: 2, 563-580, 1999.
- Chatterjee, S., S.A. Slotnick, M.J. Sobel, "Delivery Guarantees and the Interdependence of Marketing and Operations," *Production and Operations Management*, 11: 3, 393-410, 2002.
- Chen, C.-Y., Z.-Y. Zhao, M.O. Ball, "Quantity and Due Date Quoting Available to Promise," *Information Systems Frontiers*, 3: 4, 477-488, 2001.
- Cheng, T.C.E., M.C. Gupta, "Survey of Scheduling Research Involving Due Date Determination Decisions," *European Journal of Operational Research*, 38: 2, 156-166, 1989.
- Coles, S., J. Rowley, "Spreadsheet Modeling for Management Decision Making," *Industrial Management and Data Systems*, 96: 7, 17-25, 1996.
- Dhaenens-Flipo, C., G. Finke, "An Integrated Model for an Industrial Production-Distribution Problem," *IIE Transactions*, 33: 9, 705-715, 1999.
- Greene, A., "Making Better Promises," *Manufacturing Systems*, 19: 4, 22, 2001.

- Hariharan, R., P. Zipkin, "Customer-Order Information, Leadtimes, and Inventories," *Management Science*, 41: 10, 1599-1607, 1995.
- Hegedus, M.G., W. J. Hopp, "Due Date Setting with Supply Constraints in Systems Using MRP," *Computers and Industrial Engineering*, 39, 293-305, 2001.
- Hopp, W.J., M.R. Sturgis, "A Simple, Robust Leadtime-Quoting Policy," *Manufacturing and Service Operations Management*, 3: 4, 321-336, 2001.
- Hopp, W.J., M.L. Roof Sturgis, "Quoting Manufacturing Due Dates Subject to a Service Level Constraint," *IEE Transactions*, 32: 9, 771-784, 2000.
- Johnson, L.A., D.C. Montgomery, *Operations Research in Production Planning, Scheduling and Inventory Control*, John Wiley and Sons: New York, 1974.
- Katok, E., D. Ott, "Using Mixed-Integer Programming to Reduce Label Changes in the Coors Aluminum Can Plant," *Interfaces*, 30: 2, 1-12, 2000.
- Kirche, E.T., S.N. Kadipasaoglu, B.M. Khumawala, "Maximizing Supply Chain Profits with Effective Order Management: Integration of Activity-Based Costing and Theory of Constraints with Mixed-Integer Modeling," *International Journal of Production Research*, 43: 7, 1297-1311, 2005.
- LeBlanc, L.J., J.A. Hill Jr., G.W. Greenwell, A.O. Czesnat, "Nu-kote's Spreadsheet Linear-Programming Models for Optimizing Transportation," *Interfaces*, 34: 2, 139-146, 2004.
- McClelland, M.K., "Order Promising and the Master Production Schedule," *Decision Sciences*, 19: 4, 858-879, 1988.
- Moodie, D.R., P.M. Bobrowski, "Due Date Management: Negotiating the Trade-off between Price and Delivery," *International Journal of Production Research*, 37: 5, 997-1021, 1999.
- Moses, S., H. Grant, L. Gruenwald, S. Pulat, "Real-Time Due-Date Promising by Build-to-Order Environments," *International Journal of Production Research*, 42: 20, 4353-4375, 2004.
- Plenert, G., "What Does a Decision Support System (DSS) Do for Manufacturing?" *Production and Inventory Management Journal*, 33: 1, 17-19, 1992
- Smith, B.C., J.F. Leimkuhler, R.M. Darrow, "Yield Management at American Airlines," *Interfaces*, 22: 1, 8-31, 1992.
- Smith, G.A., "Insights from Industry: Using Integrated Spreadsheet Modeling for Supply Chain Analysis," *Supply Chain Management*, 8: 3/4, 285-290, 2003.

Taylor, S.G., G.J. Plenert, "Finite Capacity Planning," *Production and Inventory Management Journal*, 40: 3, 50-56, 1999.

Troutt, M.D., "Spreadsheet Usage: Some New Challenges and Opportunities," *Journal of Organizational and End User Computing*, 17: 1, I-V, 2005.

Vollman, T.E., W.L. Berri, D.C. Whybark, *Manufacturing Planning and Control Systems*, 4th ed. Irwin/McGraw-Hill: New York, 1997.